Video Temporal Segmentation using Applause Sound and End-of-Act Detection for a Circus Performance Video Archive

A thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

Lukman Hakim Iwan
Bachelor of Computer Science
Master of Computer Science

School of Computer Science and Information Technology
College of Science, Engineering and Health
RMIT University

December, 2015

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis/project is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

Lukman Hakim Iwan
December, 2015

Acknowledgments

I would like to start off my words here by giving my humble thanks to Almighty Allah for bestowing His blessings in the form of a supporting family, strength, knowledge and a very supportive group of friends and supervisors, which have enabled the completion of this research.

I would like to thank my first supervisor, Associate Professor James A. Thom, for his comments, guidance, and valuable advice. I appreciate his vast knowledge in many areas and his assistance in writing this thesis. I would like also thank to my second supervisor, Associate Professor David Carlin, for his help and support.

I thank the Circus Oz Living Archive Project team and the Circus Oz for the valuable feedback in the project.

I dearly thank my parents; my wife Diane; my daughters Lifi and Val; and my son, Salman Abdullah, whose love, caring and endless support made this entire endeavor worthwhile.

I really appreciate Bruna Pomella for her assistance in proof reading this thesis.

I acknowledge that this research would not have been possible without the opportunity and scholarship provided by the Australian Research Council (ARC) and the Royal Melbourne Institute of Technology (RMIT) University.

Credits

Portions of the material in this thesis have previously appeared in the following publications:

- Laurene Vaughan, Reuben Stanton, Lukman Iwan, Jeremy Yuille, Jane Mullet, David Carlin, James Thom, Adrian Miles. 2013. Multimodal experiments in the design of a living archive. Nordic, *Design Research Conference 2013*, Copenhagen.
- Lukman Iwan. 2013. Automatic Video Temporal Segmentation, Digital Acrobatics: Performing the Circus Oz Living Archive. RMIT University. 4-5 July.
- Lukman Iwan. 2015. Clues for Temporal Segmentation of Circus Videos into Acts in *Performing Digital: Multiple Perspectives on a Living Archive.* eds D Carlin and L Vaughan. Ashgate, Farnham UK.
- Lukman H Iwan, James A. Thom. Temporal video segmentation: Detecting the end-of-act in Circus performance video archives. *Multimedia Tools and Applications.* (to appear, accepted for publication 1 December 2015)

This work was supported under Australian Research Councils Linkage Projects funding scheme (project number LP100200118). We would like to thank our partners on the project: Australia Research Council, RMIT University, LaTrobe University, Circus Australia Ltd, Australia Council for the Arts, and Victoria Arts Centre Trust.
The thesis was written in the `Vim` editor on Mandrake GNU/Linux, and typeset using the LaTeX 2_ε document preparation system.
All trademarks are the property of their respective owners.

Note

Unless otherwise stated, all fractional results have been rounded to the displayed number of decimal figures.

Contents

List of Figures

List of Tables

Abstract

Typically, archival performance videos are: filmed in a single shot, lengthy, affected by camera operation, and originate from various video formats. To be useful, a video of a whole performance needs to be segmented into discrete acts that represent individual clips within the total performance; however, this is not a simple task due to the characteristics of the video content.

The Circus Oz video collection is an existing performance video archive that comprises over 1,074 videos totaling over 1,000 hours of viewing. To deliver their video collection to users, a prototype of the Circus Oz performance video archive system has been developed which includes system architecture and database schema.

For the purpose of video segmentation, we identify the specific clues that indicate where a performance video is likely to be segmented: that is, when an applause sound is detected in combination with one or more other clues such as black frames and image changes.

An applause detection technique for multiple applause classes has been proposed. In order to evaluate the performance of the proposed technique, an audio data set together with applause ground truth data on a sample of the Circus Oz performance videos have been developed. This applause data set contains three applause classes: less clap, more clap, and pure clap.

The proposed applause detection technique uses both characteristic-based and classification-based approaches. Our experiments show that minimum applause strength and duration values are the two essential threshold values for improving the precision of applause detection using the classification-based approach. In this approach, we found the optimum combination of several audio features. In our applause classification experiment, we achieved 83%,

94%, and 100% correctly classified for quaternary, ternary, and binary class classification respectively.

Using the clues we identified, we proposed a method for detecting end-of-act using applause sound detection, black frames detection and image comparison. The experiment shows that the precision and recall of the end-of-act-detection method is 49% and 92% respectively, making the task of manual annotators much more productive.

Chapter 1

Introduction

1.1 Background

Temporal video segmentation is not new in the video processing and analysis field. Many researchers have proposed temporal video segmentation techniques based on audio, visual or audio-visual content approaches. However, the majority of these have been applied to the segmentation of TV programs and movies into meaningful clips such as commercial clips [Liu et al., 2011; 2010], news clips [Subashini and Palanivel, 2012], and movie chapters [Chen et al., 2008; Cao et al., 2003; Chasanis et al., 2009; Hanjalic et al., 1999]. To our knowledge, no researchers have explored temporal segmentation techniques on performance video. These existing temporal video segmentation techniques cannot be applied in a straightforward way to segment a performance video, as the characteristics of audio-visual content are quite different from those of TV programs, movies and home videos.

Segmentation of archival performance videos is challenging due to their particular characteristics. The majority of videos are filmed in a single shot and are lengthy, usually running for around an hour and a half. The videos are affected by camera operation techniques such as zooming in/out and panning left/right to follow the performance on the stage. Flashlights and other dynamic stage lighting can shine directly into the camera. The quality of the audio is variable and there may be noises from the audience. The videos may originate from one of several recording formats such as u-matic, VHS, Betamax, mini-dv and digital formats.

Temporal video segmentation benefits users. They can interact directly with a specific clip of video rather than a whole long video. Users are able to retrieve short clips easier and faster than scanning through a long video. Sharing and commenting features can also be done on the clips as well as on whole videos. Furthermore, users can make a clip collection by grouping interesting clips from across different videos.

In order to deliver archival performance video to the user, a video retrieval system needs to be developed specifically for the archive. However, developing a video archive system is not the same as for a general video retrieval system. The reason is that the development of this system is focused on the handling efficiency associated with loading huge multiple video versions, maintaining the archive data, and dealing with copyright issues.

To increase its usefulness, a video of a whole performance needs to be segmented into discrete clips that represent individual clips within the performance; however, this is not a simple task due to the particular characteristics of these videos. We identify the specific clues that indicate where a performance video is likely to be segmented. The segmentation clue in the archival performance video is when an applause sound is detected in combination with one or more other clues such as black frames and image changes.

In regard to the applause detection technique, an applause data set needs to be developed to test the performance of the proposed applause detection technique. The applause detection technique has been applied to several different audio data sets including: auditory training systems [Lesser and Ellis, 2005], meeting speeches [Manoj et al., 2011; Li et al., 2009], and music concerts [Sarala et al., 2012; Marmaroli et al., 2013]. Unfortunately none of them are published.

Setting up an applause data set for a performance video archive is challenging. The reason is that typical archival performance videos are lengthy and contain quite a lot of applause on one video; moreover, it is rather difficult to find when and where the applause occurred. The applause sound could occur at the same time as other sounds or other sounds could be heard as applause. In addition, finding the exact start time and end time of the applause sound further complicates this task.

An archival performance video can be a segmented applause-based sound occurrence.

One of the essential sound clips for performance video segmentation is an applause sound as it gives a strong indication that something is interesting to, or has been acknowledged by, the audience or that something has ended. Most clapping sound detection techniques use supervised machine-learning techniques [Olajec et al., 2006; Jarina and Olajec, 2007; Cai et al., 2006]. It has been reported that audio signal characteristic-based (rule-based) techniques perform better than supervised learning [Manoj et al., 2011; Li et al., 2009]. However, the clapping sound in the dataset of meetings [Olajec et al., 2006; Manoj et al., 2011; Li et al., 2009] is usually a pure clapping sound, whilst the clapping sound in performance video archives mostly occurs at the same time as other sound types such as music, speech, audience noise, laughter and cheering. This is one of the reasons that audio signal characteristic-based techniques are unlikely to be effective for the circus performance video archive.

On the Circus Oz video, the meaningful segment is a circus act. Segmenting video into acts is challenging. Many researchers propose frameworks for segmenting video into clips, these frameworks are mostly based on shot detection techniques [Chen et al., 2008; Cao et al., 2003; Chasanis et al., 2009; Hanjalic et al., 1999]. However, video segmentation frameworks based on shot detection are not useful for the circus performance videos which were usually recorded in one single shot.

Here, we propose the methods for segmenting a performance video archive based on applause detection and image changes. In addition, as this research is part of an ARC Linkage project in conjunction with Circus Oz, the proposed techniques will be applied to the Circus Oz video archive collection. The entire video archive comprises over 1,074 videos totaling over 1,000 hours of viewing. Furthermore, a prototype of the online Circus Oz video system is developed. This prototype includes a video server, Circus Oz application, database schema, search functionality, and video processing.

1.2 Research questions

The research questions of this thesis are:

1. How can an effective and efficient performance video archive retrieval system be developed for managing lengthy videos of performances?

2. How to construct an applause data set for evaluating the proposed applause detection techniques?

3. What algorithms are effective for detecting applause sound in archival performance videos?

4. What algorithms are effective for automatically segmenting videos into meaningful clips for subsequent search?

1.3 Methodology

This section provides an overview of the methodology applied to answer these four research questions.

Video archive retrieval system prototype development. In order to develop a performance video archive retrieval system, four main components need to be explored. First, the system architecture for a performance video archive system must be analysed. Next, a database schema is developed. This is followed by the search function. Lastly, is the development of automated audio/video processing.

Applause data set development. In order to develop an applause data set, initially a video data set is selected from the Circus Oz video collection. This data set is divided into two sets: development and test sets. After that, applause classes are defined. The classes could be simply two classes: applause and non-applause, or it could be non-applause and multiple applauses classes. Next, an applause ground truth for each video is developed. This can be done by manually labeling each video to indicate where and when an applause sound occurs. Finally, the applause ground truth is saved into a database and published.

Applause sound detection. The existing techniques for detecting applause are explored. Existing techniques can be divided into two approaches: characteristic-based and classification-based approaches. The main task in the characteristic-based approach is finding a unique characteristic audio feature for the applause sound. After that, a simple rule is developed by

applying several threshold values. The main task in the classification approach is a learning process. Initially, the sound is labeled according to given classes. Next, selected audio features are extracted. Finally, the extracted audio features are submitted to a machine-learning algorithm to build a classification model of applause sound.

Detecting End-of-act in circus video. To segment a circus performance video into acts, the audio and video contents of a Circus Oz video need to be analyzed. Audio-visual analysis focuses on finding clues to when an act ends. Once the audio-visual clues have been found, we then develop a method for detecting the end-of-act on a circus performance video.

1.4 Contribution

The contributions of this research are as follows:

A prototype of video archive system. A prototype of a performance video archive system is developed including the system architecture and database schema. The system architecture is efficient enough to handle the unique characteristics of a performance video archive. The database schema is quite flexible and can therefore be implemented on different content as long as the system structure has a logical relation of video and clips.

Applause Sound Data set Although many researchers have proposed and applied applause detection techniques to various audio/video dataset, none of the datasets has been published. We publish the audio data set and ground truth of applause sound in the Circus Oz performance video archive. This applause data set contains three applause classes: less clapping, more clapping, and pure clapping.

Applause detection technique We explore two main applause detection approaches: characteristic-based and classification-based approaches. On the characteristic-based approach, we found the optimum threshold value and applause duration are essential to improve precision of applause detection technique. On the classification-based approach, we propose

a method for detecting three applause classes in the performance video archive: less clapping, more clapping, and pure clapping classes.

End-of-act detection method We propose a new method for detecting end-of-act in Circus Oz performance videos. This method consists of three methods: applause detection, black frames detection and image comparison techniques.

1.5 Thesis overview

The remainder of this thesis is organized as follows:

Chapter 2 : Literature Review

This chapter reviews existing literature related to video archives in general, applause data sets, applause sound detection techniques, and temporal video segmentation.

Chapter 3 : Circus Oz Video Retrieval System

This chapter describes the general system architecture of Circus Oz Living Archive application including: video server, Circus Oz application, database schema, search functionality and audio/video processing.

Chapter 4 : Circus Oz Dataset

The applause sound data set is explained in this chapter. The manual labeling process is described. The applause statistic ground truth is also presented in this chapter.

Chapter 5 : Applause Sound Detection

This chapter describes the technique for detecting applause sound in a Circus Oz performance video archive. A number of audio features will be explored. Furthermore, two applause detection approaches will be examined: characteristic-based and classification-based approaches.

Chapter 6 : Detecting End-of-act on Circus Performance Video Archive

The method for detecting end-of-act will be explained in this chapter. Audio-visual content is analyzed to find a strong clue that an act has ended.

Chapter 7 : Conclusion

Chapter 7 draws the conclusions of this thesis. Future work is also outlined in this chapter.

Chapter 2

Literature Review

This chapter reviews existing literature related to video archive systems, applause data sets, audio features, applause sound detection techniques, and video temporal segmentation.

More and more video archives available on the web are accessible to public users. When developing a video archive system, a number of factors need to be considered including: system architecture, system requirements, and features. Most video archives are stored in old analog format and need to be converted into digital format so that they can be accessed through the web. Another issue is the typical duration of video archives that can last for an hour or more. This issue can be tackled by segmenting the video into shorter video clips; for example, a large new video archive collection could be segmented into short topic clips (Ide et al. [2004]).

Applause detection is one means of segmenting a long archive video. There are two approaches to detecting applause: by characteristics and by classification. An applause data set is needed to test the performance of the applause detection technique. Although many researchers have developed applause data sets from various video collections, many of them are not available to public users. The key aspect of detecting applause is an audio feature enabling us to distinguish the applause sound from other sounds. Selecting suitable audio features and pre-processing them is also important for detecting the applause sound.

Video temporal segmentation has attracted a great deal of research attention, although the focus is still on segmenting movies and collections of TV programs. The videos can be

segmented based on their audio or visual contents, or both. For audio-based segmentation, specific sounds such as applause and cheering are used to split the video event. For visual-based segmentation, image comparison techniques are employed to find the boundary of the video segment.

2.1 Video archive systems

In order to construct a video archive system, three things need to be considered: conversion from analog to digital format, segmentation of long videos, as well as technical features, requirements, and system architecture.

First, the video is converted from analog to digital format. The AURORA (Automated Restoration of Original video and film Archives) system developed by van Roosmalen et al. [1998] provides a restoration system that can improve visual quality and coding efficiency before it is transferred to new storage. The AURORA system can improve the visual quality by reducing noise, blotches and intensity flicker.

Second, video segmentation is needed because a typical video in an archive is lengthy, with one or two hours'duration. Automatic video segmentation would be preferable to manual segmentation to achieve time and resource efficiency. However, both techniques can be used together; for example, automatic and manual segmentation techniques can be combined to annotate and browse a large collection of internal family movies [Abowd et al., 2003].

Various methods have been proposed to automatically segment long videos based on different video contents such as topic, tracking, sound, image and text. For topic-based segmentation, Ide et al. [2004] proposed a method for structuring a huge news video archive. The news video is segmented into topic units based on topic interest and then the related segments are linked. For tracking-based segmentation, [Ide et al., 2005]] proposed a method for exploring a large news video archive collection based on human relation in the video. They built an interface called trackThem to retrieve and track down the video content relations.

For segmentation based on sound events, Zhang and Ellis [2001] used the cheering sound to detect events in a basketball game. They found it challenging to detect specific sounds such as shooting, ball bouncing, and dribbling because those sounds are harder to distinguish

than the cheering, the commentator's voice, and the umpire's whistle in a basketball game.

Furthermore, image and text content of videos are used to visualize the semantic structure of a news video archive [Mo et al., 2004]. They proposed a technique to extract the key images from a news video archive. Text recognition in video captions is used for topic segmentation and topic threading, whereas the similarity of images is used detect identical video shots.

Lastly, there are important technical aspects to consider when developing a video archive system. These include: key features, system requirements, and system architecture.

Hjelsvold et al. [1995] argue that there are four main key features to be considered when developing a video archive: searching, structure browsing, contents browsing, and video annotation. Search is a feature for finding a short segment from the video archive and other related short segments should also be displayed in the search results. Structure browsing will help users to quickly scan the video archive rather than watching long, entire videos. Video content metadata added to the database allows users to do contents browsing such as the people who appear in a video or the place and time when a video was recorded. Video annotation makes video searching and browsing tasks easier. In addition, effectiveness of video retrieval technology to the contents of an historical video archive has been studied [Petrelli and Auld, 2008]. They found that most users choose searching functionality rather than browsing functionality. However, they still use the browsing functionality when they are not happy with the search results.

Zou et al. [2001] argue that a video archive system has to meet the following system requirements: software must re-usable, multimedia content has to be dynamically kept up to date, modular programming, structured media collection, and unified language system. Furthermore, their video platform has the following features: data collection accessible to several user levels, multiple relational tables, video and metadata are searchable, and a flexible data representation.

Trifonov and Georgieva [2009] propose a client/server system architecture for clips from an audio and video archive of Bulgarian bells. Their system architecture consists of three layers as shown in Figure 2.1. They are: data processing, application, and client layers. On the server side, the data processing layer is applied using a database management system,

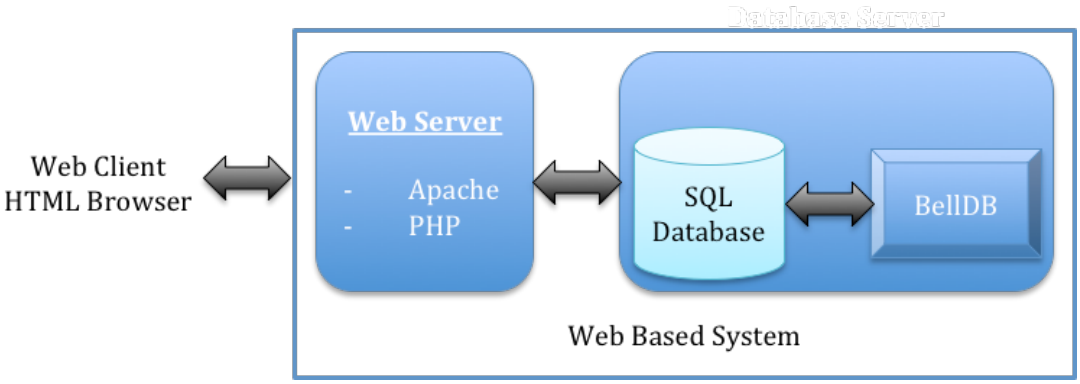


Figure 2.1: Client/server video archive system architecture for unique Bulgarian bells

while the application layer is implemented by using a web server. On the client side, a Web client HTML browser is used.

2.2 Applause data sets

Applause is one of the most interesting audio keys for indexing audio recordings. Applause sound detection has been applied in at least three areas: audio segmentation, audio key detection, and audio classification. The most common issue associated with audio segmentation is finding potential audio clues. Based on these clues, the audio can be segmented into parts. One example is segmenting the Montreux Jazz Festival Concerts into tracks [Marmaroli et al., 2013]. Another example is finding the end of a song in archival Carnatic music [Sarala et al., 2012]. They both use the applause sound as a clue that the song has ended.

Next is audio key detection. Audio key is the most useful information to index audio data. One example of audio key is the applause sound. This detected type of sound is a strong indication that something interesting occurred on that audio track. Li et al. [2009] and Manoj et al. [2011] use the detected applause sound to highlight interesting events in video. The detected applause and laugh sounds are used for highlight interesting events in video [Shi et al., 2011]. The detected applause sound is also used as a key audio effect [Cai et al., 2003; 2006].

Last is audio classification. The sound of applause is one of the most interesting sounds in audio classification, where the data set usually contains short audio clips. These clips

are then categorised into one or more classes. Uhle [2011] categorised the clips, taken from commercial recordings of music and speech, into six applause classes. Clips taken from TV and radio are also categorised into two classes: applause and laugh [Pinquier et al., 2004].

Furthermore, a general framework for audio classification has been proposed ([Harb and Chen, 2007]). This general framework is not problem specific; it can be used for any audio classification problem such as: speech/music discrimination, gender recognition, highlights detection and musical genre recognition.

Applause detection techniques have been applied to several audio data collections including auditory training systems, TV programs, radio broadcasts, movies and live music performances. Auditory training system audio data is used for detecting near-field clapping and far-field clapping [Lesser and Ellis, 2005]. The applause detection technique has also been applied to audio data of meeting speech [Manoj et al., 2011; Li et al., 2009]. A TV program audio collection has also been explored to detect applause sound [Olajec et al., 2006; Jarina and Olajec, 2007]. Furthermore, the Montreux jazz festival concert is segmented into tracks by detecting the applause sound at the end of the concert [Marmaroli et al., 2013] and an archival Carnatic music audio recording has been segmented into a sequence of pieces using the characteristics of applause and music [Sarala et al., 2012].

The statistics of audio data sets that have been explored for detecting applause are shown in Table 2.1. Unfortunately, these audio data sets have not been published.

Detecting the sound of applause on audio is quite often not only focused on finding the applause sound itself, but also finding other key sound types. In fact, the content of audio data is a combination of several sound types such as music, speech, silence, and laughter. Determining sound type in an audio data collection for detecting the applause sound is an important task. This task involves distinguishing the applause sound from other sounds There are other interesting key sounds, such as laughter and speech, for indexing audio streams:. The statistics of applause data sets that have a different number of classes are shown in Table 2.1.

Most researchers detect the applause and non-applause sounds in their audio data by using characteristic-based approaches. This is because they are focusing on analyzing the

Papers	Category	Length	# classes	# applause classes
Cai et al. [2003]	TV programs	2hours	3	1
Pinquier et al. [2004]	TV programs	12 hours	4	1
Lesser and Ellis [2005]	Audio	1,650 claps	2	2
Olajec et al. [2006]	TV programs	6.5 hours	2	1
Jarina and Olajec [2007]	TV programs, music	9 hours	2	1
Li et al. [2009]	speech	50 hours	2	1
Shi et al. [2011]	mix	20 hours	2	1
Uhle [2011]	music, speech	210 clips	6	5
Manoj et al. [2011]	speech	43 hours	2	1
Sarala et al. [2012]	music	19 concert	2	1
Marmaroli et al. [2013]	music	5,000 hours	2	1

Table 2.1: Applause data set statistics

unique characteristics of the applause sound. Applause and non-applause sounds are detected in order to determine the end of songs in music concerts [Marmaroli et al., 2013; Sarala et al., 2012]. Using the classification-based approach, Olajec et al. [2006] and Jarina and Olajec [2007] classified an audio clip into two classes: applause and non-applause.

Other researchers are interested in applause and non-applause as well as key audio effect classes. This is mostly about sound key detection in audio streams with classification-based approaches. Applause and laughter classes are defined by Shi et al. [2011] and Pinquier et al. [2004]. The key sound effects defined by Cai et al. [2006] include applause, car-racing, car-crash, cheers and explosion.

Furthermore, the applause class itself can be further divided into a number of applause classes. This is useful if the audio data contain different types of applause. Lesser and Ellis [2005] divided applause classes into near-field and far-field clapping, while Uhle [2011] defined six different classes ranging from no-applause to pure applause.

2.3 Audio features

Choosing audio features is the most important part of an applause detection technique. These features can be used for distinguishing between an applause sound and a non-applause sound. Many researchers suggest some useful audio features for the automatic audio classification

Audio Features	Characteristic based	GMM (Gaussian Mixture Models)	RLSC (Regularized Least-Squares Classifier)	HMM (Hidden Markov Model)	Multilayer
MFCC (Mel Frequency Central Coefficients)		Olajec et al. [2006]; Jarina and Olajec [2007]		Cai et al. [2006]	Uhle [2011]
Spec. Grav. Center	Marmaroli et al. [2013]				Uhle [2011]
Spec. Roll-off	Marmaroli et al. [2013]				Uhle [2011]
Temporal	Marmaroli et al. [2013]				
Spec. Flux	Sarala et al. [2012]	Pinquier et al. [2004]		Cai et al. [2006]	Uhle [2011]
Spec. Entropy	Sarala et al. [2012]				Uhle [2011]
Center Mass			Lesser and Ellis [2005]		
Slope			Lesser and Ellis [2005]		
Cross Corellation			Lesser and Ellis [2005]		
Energy			Lesser and Ellis [2005]	Cai et al. [2006; 2003]	
ZCR (Zero Crossing Rate)			Lesser and Ellis [2005]	Cai et al. [2006; 2003]	
Brighness				Cai et al. [2006; 2003]	
Bandwidth				Cai et al. [2006; 2003]	
Harmonic prominence				Cai et al. [2006]	
Spares Rep.				Shi et al. [2011]	
Duration	Li et al. [2009]				
Pitch	Li et al. [2009]				
Spectrogram	Li et al. [2009]				
Occurrence location	Li et al. [2009]				
Amplitude	Manoj et al. [2011]				
Lag Value	Manoj et al. [2011]				

Table 2.2: Audio features set and detection technique for applause sound

task, such as mel-frequency central coefficients (MFCC) [Briggs et al., 2009; Subashini and Palanivel, 2012; Silva, 2012; Lu et al., 2003], spectral [Briggs et al., 2009; Silva, 2012; Lu et al., 2003; McKay, 2005; Zhang and Kuo, 1999; McKay, 2010], frequency [Briggs et al., 2009; McKay, 2005; Zhang and Kuo, 1999; Kiranyaz et al., 2006; Theodorou et al., 2012], zero crossings [Silva, 2012; Lu et al., 2003; McKay, 2005; Zhang and Kuo, 1999], and energy [Lu et al., 2003; McKay, 2005; Zhang and Kuo, 1999; Kiranyaz et al., 2006; Theodorou et al., 2012; Lesser and Ellis, 2005]. In particular, MFCC is combined with a color histogram for finding a transition point between news and advertisements in TV programs [Subashini and Palanivel, 2012]. Spectral centroid, spectral roll-off and spectral flux features are selected for classifying cinematic sound [Silva, 2012]. To build a generic audio classification and segmentation, fundamental frequency and sub-band centroid frequency estimation is used in Kiranyaz et al. [2006]. The energy feature is one of the audio features selected for detecting the sound of applause [Lesser and Ellis, 2005]. The zero-crossing rate feature is used for classifying various audio types in radio news broadcasts [Theodorou et al., 2012].

Based on literature review, the following audio features are suitable to detect applause sound:

Spectral Flux

Spectral flux calculates how much the spectral changes from one frame to another frame [McKay, 2010]. Formula for calculating the Spectral Flux (SF) at time t is as follows:

$$SF_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2 \tag{2.1}$$

where $N_t[n]$ = normalized magnitude spectrum.

Spectral Roll-Off

Spectral Roll-Off calculates the energy of the signal below the lower frequencies. Typically, the lower frequency limit is less than 0.85 or 0.95 [McKay, 2010]. The formula for calculating

the Spectral Roll-Off, SR_t, is as follows:

$$\sum_{n=1}^{SR_t} P_t[n] = k \sum_{n=1}^{N} P_t[n] \tag{2.2}$$

where:

$P_t[n]$ = power spectrum

k = lower frequency limit

2.3.1 Audio feature selection

In a classification-based approach, the smaller number of features submitted to the classifier, the more efficient is the computation time. Too many features submitted to a classifier do not necessarily mean a better performance. The reason is that one or more features may have an insignificant or no effect on improving performance. One way to resolve this issue is by carefully selecting the features submitted to the classifier, and removing a feature if it has an insignificant effect on the performance, and add the feature that have a large effect.

For audio feature selection, Olajec et al. [2006] and Jarina and Olajec [2007] analyze which MFCC coefficient has a large effect on distinguishing the applause sound from other sounds. They used used GA and SA tools to find the best MFCC coefficients on distinguishing applause sound and other sound. They found that MFCC coefficients and their derivatives play an important role on detecting applause sound. For example, MFCC coefficient 1 and coefficient 24 perform best [Olajec et al., 2006]. Jarina and Olajec [2007] found that selecting MFCC 1st, 2nd, 3rd perform best using GA and SA feature selection.

Furthermore, Uhle [2011] discards the audio features that do not contribute to the classification process, using backward selection for the pruning. The set features are removed one by one until the performance declines.

MFCCs coefficient selection

MFCC is one of the audio features that are widely used in speech recognition and applause detection. At least two researchers have undertaken some research on the selection of a suitable audio feature for clapping sound detection. Olajec et al. [2006] used the classical technique for speech recognition (MFCCs + GMM), and then selected suitable audio features for applause detection. Specifically, the properties of various MFCC coefficients are investigated. They reported that if only some coefficients are selected, the performance increased by about 18.2%. The selected audio features are explored by a genetic algorithm.

On the other hand, Jarina and Olajec [2007] investigated audio feature selection for sound detection. The audio feature was selected using Genetic Algorithm and Simulated Annealing. The result is quite similar to that obtained by Olajec et al. [2006] who concluded that only MFCCs improve the performance. The difference is that Jarina and Olajec [2007] conducted further analysis by comparing MFCC and their derivatives. They conclude that the delta-delta coefficient highly outperforms the delta coefficients.

2.3.2 Feature extraction

Another important process in the investigation of audio features is feature extraction where two important things must be considered: audio format and window/frame size. In feature extraction, the audio format is usually mono with 16 bits. Olajec et al. [2006] and Jarina and Olajec [2007] use 22.050 Hz audio frame while Uhle [2011] use 44.100 Hz audio format.Extracting an audio feature is not just a matter of extracting the feature for whole clips. We normally extract short frames of audio. Technically, the shorter frames the better. The normal frame ranges from 20 milliseconds to 40 milliseconds. Uhle [2011] and Olajec et al. [2006] extracted audio frame 23ms with 50% overlap.

There are at least two audio tools for extraction audio features: jAudio and opensmile. jAudio is software for audio extraction developed by McEnnis et al. [2005]. The features include: spectral, rms, energy, beat, MFCC, and LPC. The resulted features can be saved into weka file format (arff) or xml files. Similarly, opensmile is also built to extract audio features. The audio features include: energy, spectral, MFCC and PLP.

2.3.3 Feature post processing

In order to obtain an optimal applause detection result, the original audio feature value is usually normalized and smoothed. Normalization is necessary as the feature value range could be wide. The audio signal quite often fluctuates extremely, so it needs to be smoothed by filtering the signal with Finite Impulse Response (FIR) or Infinite Impulse Response (IIR).

For audio feature normalization, Sarala et al. [2012] compare un–normalized and normalized audio spectral flux to detect applause in a Carnatic music concert. The performance of their proposed algorithm with peak-normalization spectral flux is better than with un-normalized spectral flux. Furthermore, Sarala et al. [2012] applied the CUSUM formula after normalizing their audio feature set. The CUSUM value determines the strength and duration of the applause sound.

Uhle [2011] smoothed and normalized the audio features before submitting them to the classifier. The audio features were normalized using mean and variance, and smoothed using FIR and IIR filters. Their experiment result showed that the IIR filter performed better than did the IIR filter.

2.4 Applause sound detection techniques

This section highlights the state-of-the-art applause sound detection technique. This includes applications for applause detection, audio feature selection techniques, and applause detection techniques. In general, there are two approaches for detecting the sound of applause: the characteristic approach and the classification approach. The former analyzes the different characteristics of particular audio features. Based on that, the rule for detecting applause is developed by applying several threshold values.

The classification approach takes the applause detection task into the data-mining field. Initially, potential sets of audio features are explored. After that, audio features are selected that contribute the most to distinguishing the applause sound from other sounds. The selected audio features are then submitted to machine learning tools such as Weka [Hall et al., 2009]. Mostly, they use the supervised machine learning approach where a set of training data is defined and labeled manually. Finally, an applause sound classification model is built

using one of the classifiers in the Weka data mining tool such as:

- BayesNet (Bayesian Network): a comprehensive graphical representation of probability distribution.
- MLP (Multilayer Perceptron): a feed-forward neural network that has multiple layers: input layer, hidden layer(s), and output layer.
- Decision Tree-J48: an implementation of C4.5 algorithm (extention of ID3 algorithm)
- SVM-SMO (Support Vector Machines - Sequential Minimal Optimization): supervised learning methods for classification and regression using sequential minimal optimization.
- FT (Functional Trees): a classifier for building functional trees.
- SimpleLogistic: a logistic regression with only one parameter.

Both characteristic and classification approaches have been explored by quite a few researchers as shown in Table 2.3.

2.4.1 Characteristic approach

In the characteristic approach, the properties of the applause sound, including audio features, duration, and occurrence location, are carefully analysed. The audio features used in the characteristic-based approach are not as many as in the classification approach which usually uses more audio features. Based on that analysis, a simple rule is developed which contains threshold values for each audio feature.

In distinguishing applause from speech in the audio recording of meeting speech, the audio features are analyzed to find the characteristic difference between the two [Li et al., 2009]. These audio features include: duration, pitch, spectrogram, and occurrence location. Based on the analysis of these different characteristics, the rule for splitting the two sounds is established. Applause is detected if the clip duration is greater than 0.8 second and less than 3.6 seconds, while speech is detected if the clip duration is less than 3 seconds.

Detection Approach	MFCCs	Spectral	Spectal, Temporal	MFCC, Spectral, Temporal	Others
Characteristic:					
		Sarala et al. [2012]	Marmaroli et al. [2013]		Li et al. [2009]; Manoj et al. [2011]
Classification:					
GMM	Olajec et al. [2006]; Jarina and Olajec [2007]	Pinquier et al. [2004]			
RLSC					Lesser and Ellis [2005]
HMM				Cai et al. [2003; 2006]	
hMM					Shi et al. [2011]
Multilayer Perceptron				Cai et al. [2003]	

Table 2.3: Applause detection approaches

Pitch of applause is usually 0, while pitch of speech is between 50 Hz and 800 Hz. The spectrogram analysis shows that the applause is non-spectral, while the speech clip is spectral. Furthermore, the applause mostly occurred before or after speech. There is no overlap between applause and speech. The data flow for detecting applause sound is presented in Figure 2.2.

Sarala et al. [2012] proposed an applause detection technique based on the characteristic approach. They analyzed the signal changes between music and applause on a Carnatic music recording. Specifically, they analyzed the changing of spectral flux (SF) and spectral entropy (SE) audio signal. The spectral flux and spectral entropy are further processed with filters and normalized. Then, the CUSUM formula is applied to the normalized value to detect the level of the applause strength and its duration. From this calculation, the applause sound will be detected if the CUSUM value is greater than 0, while other sounds will be detected if the CUSUM value is equal to 0.

The CUSUM value, $Cusum[i]$, can be calculated as defined in Sarala et al. [2012] as

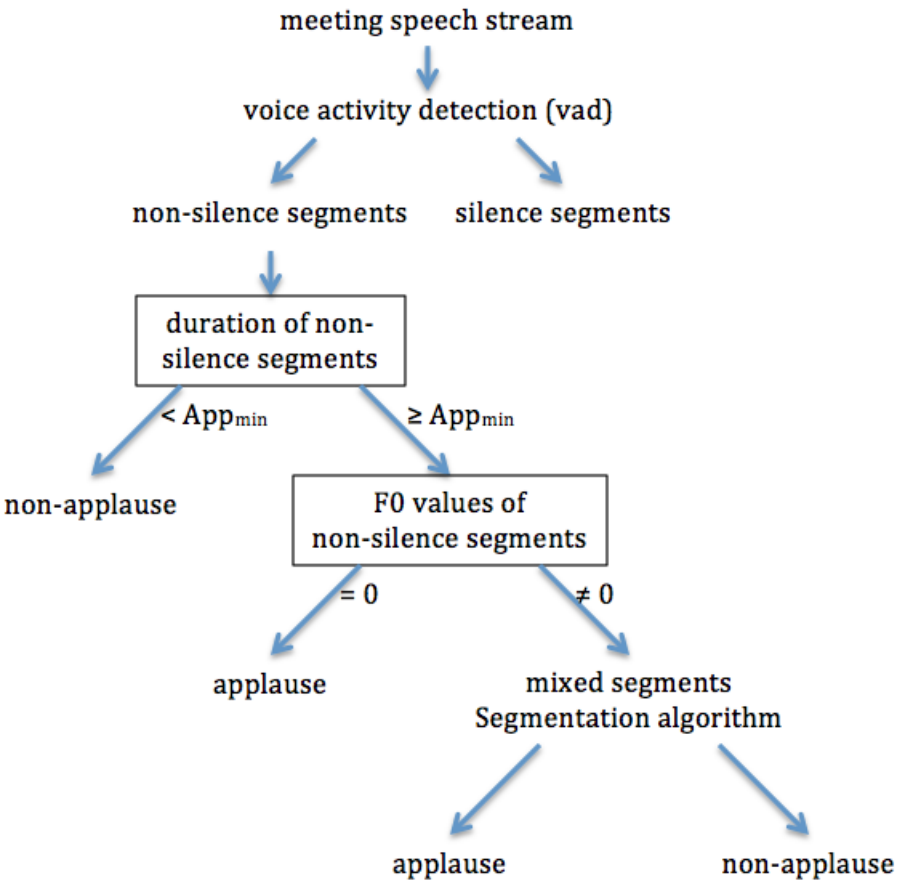


Figure 2.2: The data flow on distinguishing applause sound from speech in meetings

follows:

$$Y[i] = X[i] - \alpha \tag{2.3}$$

$$Cusum[i] = \begin{cases} Cusum[i-1] + Y[i], & \text{if } Y[i] > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

where $X[i]$ is the value of spectral flux or spectral entropy time series at time i and α is 1.5 times the average of spectral flux or spectral entropy.

Similarly, Manoj et al. [2011] used the characteristic approach to detect applause in a meeting speech recording. The characteristics of four audio features are analyzed: decay factor, lag of first minima of ACF, index of FFT bins for BER, and band energy ration. In their experiment, the set of rules was developed for the detection of applause by applying a decision threshold range to each feature. The applause sound is detected if it matches the following decision threshold ranges:

- Decay factor: 0.6 – 0.8
- Lag of first minima of ACF: 4 – 7
- Index of FFT bins for BER: 5 – 20
- Band energy ratio: 0.08 – 0.31

2.4.2 Classification approach

In the classification approach, the applause detection task is brought into the data mining area. Specifically, the audio stream is split into given clip lengths and then each clip is classified according to a given sound class. In applause sound classification, usually the supervised data mining approach is used whereby the model is constructed from a set of training data and then the model is used to detect the class of each sound clip in the audio data set. In addition, the problem and solution in multimedia data mining have been presented in [Bhatt and Kankanhalli, 2011].

In general, there are three main areas that need to be considered: audio features, pre-segmented audio stream method, and the classifier itself.

First, in the classification approach, the number of audio features used is quite often larger than that used in the characteristic approach. For example, Uhle [2011] extracted many audio features including 20 LLD, 17 LPC, 17 LSF, 12 MFCC and 12 PLP. Furthermore, Cai et al. [2006] used seven sets of audio features: MFCC, spectral flux, energy, ZCR, brightness, bandwidth and harmonic prominence features.

The audio features that are widely used in existing applause detection techniques are MFCC, spectral, energy and PLP. MFCCs are the most widely used of these techniques. This set of features is used by Cai et al. [2006], Uhle [2011], Olajec et al. [2006], and Jarina and Olajec [2007]. Another audio feature is spectral used in Briggs et al. [2009], Lu et al. [2003], and Silva [2012]. Some researchers use an energy audio feature [Lesser and Ellis, 2005; Cai et al., 2003; 2006]. The PLP audio feature is the least used [Uhle, 2011].

Second, unlike audio clip classification, applause detection in an audio stream not only classifies the clips into one of the defined classes, but also finds when that clip occurred in the audio stream. So, the pre-segmentation of the audio stream into clips is quite important. Manoj et al. [2011] and Cai et al. [2003] pre-segmented the audio stream into fixed clip lengths: 1-second and 2-second clips respectively. On the other hand, Li et al. [2009] pre-segmented the audio stream dynamically by splitting the audio stream into silence and non-silence clips. The silence audio is detected by setting up a threshold value for STE and ZCR features. After that, only the non-silence clips proceed to the classification step.

Last, the final step in the classification approach is classifying each audio clip into a defined applause class. In this step, the selected audio feature is submitted to a machine learning tool such as Weka [Hall et al., 2009]. The choice of classifier is also important in order to obtain the best applause detection performance. Cai et al. [2003] and Cai et al. [2006] employed HMM classifiers to develop a sound effect model for detecting applause, laughter, and cheering audio clips. Shi et al. [2011] used a hetMM-based classifier to detect applause and laughter in an audio clip. Their experiment shows that the performance of a hetMM-based classifier is comparable to a SVM classifier. Uhle [2011] compared several classifiers

to test the performance of the applause detection technique. Their experiment indicated that a multilayer perceptron performs better than the SVM classifier. Furthermore, MFCC features are used and neural network classification is employed to detect the cheering sound in a basketball game [Zhang and Ellis, 2001].

2.4.3 Contribution of existing applause detection techniques

This section describes the contribution of existing techniques to the applause detection area. Cai et al. [2006] proposed the detection technique for the sound of applause. In their evaluation, they compared the proposed method with classic sliding windows. Their proposed method substantially improved the recall and precision of applause detection. The recall increased from 73.2% to 97.7% and the precision increased from 47.1% to 91.4%.

Regarding applause detection evaluation, the type of data set has a unique applause content type. Cai et al. [2003] proposed a method for detecting applause which they applied to different data sets. The evaluation indicated that their method achieved better results on a sport program data set compared to a movie data set.

For audio feature selection, Jarina and Olajec [2007] and Olajec et al. [2006] analyzed which MFCC coefficient had a significant effect on distinguishing the applause sound from other sounds. They used GA and SA tools to select the best MFCC coefficients. On the evaluation, they both concluded that selected MFCC coefficients perform better. For example, MFCC coefficient 1 and coefficient 24 perform best [Olajec et al., 2006]. Jarina and Olajec [2007] found that selecting MFCC 1st, 2nd, 3rd perform best when using GA and SA feature selection.

Sarala et al. [2012] compare spectral flux and spectral entropy in distinguishing the sound of applause from music. On evaluation, the spectral entropy with peak normalization performs best. In addition, Marmaroli et al. [2013] improved the performance of applause detection by using three audio features: spectral gravity central, roll off, and temporal. After comparing the statistical approach with the characteristic approach, Manoj et al. [2011] recommended the characteristic approach. In their experiment, their proposed characteristic approach outperformed the GMM approach by 13.25%. Similarly, Li et al. [2009] proposed

the characteristic approach and compared their approach with combination of MFCC audio feature and HMM classifiers. Their evaluation indicated that the proposed characteristic approach outperformed the traditional approach by 3.65%.

2.5 Video temporal segmentation

Temporal video segmentation techniques have been studied widely and the field has focused largely on segmenting movies and TV programs. In movies, video segmentation techniques are frequently used for segmenting movies into meaningful movie scenes [Chen et al., 2008; Cao et al., 2003; Chasanis et al., 2009; Hanjalic et al., 1999]. In TV programs, segmentation techniques are usually applied to find TV commercial segments within TV programs [Liu et al., 2011; 2010; Duan et al., 2008].

The majority of approaches use shot detection techniques to find an initial segmentation point based on visual features [Liu et al., 2012; Yuan et al., 2007; Kijak et al., 2006]. Once an initial point has been identified, each shot is further analysed in order to obtain a more precise segmentation point or to merge the shots into one scene or chapter [Chen et al., 2008; Chasanis et al., 2009; Sundaram and Shih-Fu, 2000]. However, these shot-detection segmentation techniques cannot be applied directly to most performance video archives since a performance video has characteristics that are different from those of a TV program or movie collection. Therefore, we need to analyse the audio-visual content differently.

Automatic video segmentation techniques can be classified into three main content-based approaches: visual, audio, and audio-visual. Most techniques are based on a visual content approach [Liu et al., 2012; Yuan et al., 2007], while some techniques use an audio features approach [Cao et al., 2003; Silva, 2012; Zhang and Kuo, 1999; Kiranyaz et al., 2006]. The effectiveness of video segmentation techniques is reported as being more accurate when both audio and visual features are used [Covell et al., 2006; Duan et al., 2006; Lienhart et al., 1999; Sidiropoulos, 2011].

One visual segmentation technique proposed by Lienhart et al. [1997] and Sadlier et al. [2001] uses black frames to segment the video. Black and silent frames analysis is frequently used to detect advertising breaks within TV programs.

Another visual segmentation approach is image comparison. There are several techniques for comparing images including pixel-based, color histogram and comparison of detected edges [Yuan et al., 2007]. The invariance and the sensitivity of these image comparisons can be compared: the pixel-based comparison is the most sensitive method whereas the color histogram is less sensitive and more invariant than the pixel-based comparison. The edge detector comparison rarely outperforms the color histogram and is computationally expensive.

Audio classification is another approach to video temporal segmentation and many researchers suggest useful audio features for this task [Silva, 2012; Kiranyaz et al., 2006; Subashini and Palanivel, 2012; Theodorou et al., 2012; Lesser and Ellis, 2005]. The application of audio classification techniques can be extended to detecting particular sounds of interest in video, such as the sound of applause [Zhang and Kuo, 1999; Lesser and Ellis, 2005].

Many researchers have proposed methods for segmenting video based on both audio and video features [Subashini and Palanivel, 2012; Subashini et al., 2011; Sundaram and Shih-Fu, 2000; Gillet and Richard, 2006; Ceillet et al., 2014]. In particular, colour histogram and MFCC features are used for television programs [Subashini and Palanivel, 2012; Subashini et al., 2011] and music video [Gillet and Richard, 2006] segmentation. Several audio features (zero crossing rate, low energy fraction, and spectra) combined with a colour histogram have been used to segment commercial films [Sundaram and Shih-Fu, 2000]. Video contents analysis is an important initial step on video segmentation. All video content sources (visual, audio, and text) should be integrated in the segmentation process ([Snoek and Worring, 2005]). To make video content analysis effective, all the video content sources need to be converted into standard form.

For image-based segmentation, several researchers have used black frame detection techniques to segment video into scenes. Black and silent frames are used for detecting TV commercials [Lienhart et al., 1997].

Furthermore, image comparison techniques have been explored. There are several techniques for comparing images including pixel-based, colour histogram and detected edge comparison. The invariance and the sensitivity of various image comparisons have been con-

sidered [Yuan et al., 2007]. The pixel-based comparison is the most sensitive method. The colour histogram is less sensitive and more invariant than the pixel-based comparison. The edge detector comparison rarely outperforms the colour histogram and it is computationally expensive. Furthermore, unsupervised clustering of the colour histogram similarity technique has been proposed for detecting camera shots on TV program classification [Günsel et al., 1998].

Video segmentation techniques have been used in many practical applications such as: medical imaging, computer-guided surgery, machine vision, object recognition, surveillance, content-based browsing, and augmented reality application [Ngan and Li, 2011]. For example: camera software that can predict behavior, Cromatica, has been tested at London's Liverpool Street station [Wakefield, 2002]. In content-based browsing, applause detection technique has been used to split the Montreux jazz festival concert [Marmaroli et al., 2013] and Carnatic music audio recordings [Sarala et al., 2012] into tracks.

2.6 Summary

In this chapter, we have presented a review of the literature related to video archive systems, applause detection, and video segmentation techniques.

Video archive systems There are three things that need to be considered when developing a video archive system: video conversion, video segmentation, and system architecture. All three are taken into account when developing the Circus Oz video archive system as described in Chapter 3.

Applause data set The applause detection technique has been implemented for various tasks: audio segmentation, audio key detection, and audio classification. We use applause detection to classify audio content in the Circus Oz videos as described in Chapter 5.

Furthermore, the applause detection technique has been applied to several types of audio data collection contexts including: auditory training systems, TV programs, radio broadcasts, movies and live music performances. We develop an applause data set and its ground truth

from the Circus Oz video archive as described in Chapter 4.

Different audio classes have been set up for different purposes. On the one hand, some researchers have divided the audio classes into applause and applause classes as they are focusing on finding applause sound. On the other hand, others have divided the audio classes into a number of key audio effects including: applause, cheering, laughter, and speech. In addition, some researchers now focus on different type of applause classes. In this thesis, we divide the audio into different types of applause such as: less, more, and pure applause classes (Chapter 5). For video segmentation purposes (Chapter 6), we are interested in finding only the applause sound. In this case, we divide the sound into two classes: applause and non-applause classes.

Audio features In order to locate a particular sound, a number of audio features are compared. A particular audio feature can be distinguished by a particular sound. For the detection of applause, the following audio features are useful: MFCC, spectral, PLP, and energy. We use those audio features for audio classification task described in Chapter 5 and Chapter 6.

The audio format and window/frame size are two important parameters of audio feature extraction process. The audio format is usually mono with 16 bits, 44.100 Hz and the frame size range is between 20ms to 40ms. We extract the audio feature with those parameters using audio tools extraction: jAudio and opensmile (Chapter 5 and Chapter 6).

In the post-processing step, the values of audio features need to be normalized and smoothed. The normalized value can be done using normalization formula while FIR and IIR filters smooth the audio features.

Applause detection techniques The state-of-the-art applause detection technique has been highlighted. This technique can be divided into two major approaches: characteristics and classification approaches. We use both approaches to detect different types of applause sound as described in Chapter 5.

Video temporal segmentation The video segmentation technique can be classified into three main content-based aspects: audio, video, and audio-visual. We use both audio and visual content to segment Circus Oz video into acts (Chapter 5). We initially use applause to detect segment points and then conduct image comparisons.

Chapter 3

Circus Oz Video Retrieval System

The first research question of this thesis is how can an effective and efficient performance video archive retrieval system be developed for managing lengthy videos of performances? The development of a video retrieval system is not only the main objective of the Living Archive Circus Oz project, but also provides a basis for addressing the other research questions in this thesis. This chapter describes the architecture and main components of the Circus Oz video retrieval system. The system consists of a video server, an application server, database schema, search function and video processing. The application server includes a web-based front-end application although that is not a focus of this thesis.

This video retrieval system is intended for multiple audiences including academic researchers, circus performers, and public users. Academic scholars can benefit from this system by analyzing the audio and video content of the Circus Oz video collection. Circus performers will be able to access this site in order to learn various circus tricks from the past and the present videos, giving them the opportunity to improve their skills or develop a new act. The general public can watch the circus video simply to entertain themselves and their children.

A significant challenge in developing the Circus Oz video retrieval system is the nature of the videos in the collection itself, in particular the long duration of the videos, the variety of video formats, and the inconsistent video quality. Another challenge is that although the video collection should have easy and secure access, most users are not permitted access to

all videos. Therefore, our focus in developing the Circus Oz video system is to provide secure video content, efficient disk space, easy access, fast video delivery, flexible disk location, and an efficient database schema.

Firstly, secure video content is one of the main concerns as the archive video collection needs to be secure from potential hacking, and many of the videos have restrictions on who can access the content. For example, several current circus performers do not want some of their acts to be available on publically accessible videos. Music is also a major factor to be considered as some of the music in the shows have copyright issues. So the challenge here is how to handle or control a particular audio or video segment in a lengthy circus video.

A second concern is the efficiency of disk usage. As we are dealing with a large video collection, this efficiency is crucial for the video system. Users tend to watch just a short segment rather than an entire lengthy video for various reasons: they may want simple, fast access or they may be interested in a short, funny clip only. During an evaluation while developing Circus Oz video system, we asked approximately 10 users to watch a Circus Oz show video. However, most of the users did not watch the whole shows. They search for particular short act clips on that video instead. The simple solution would be to segment the video into various short clips according to specific acts, music, or something that is useful or of interest to users. However, this will produce overlaps of duplicate segments which will cause a problem with disk capacity.

The third consideration is fast video delivery. As the Circus Oz archive contains over a thousand videos that can be accessed through the Internet, fast video delivery is another main requirement for this system. Users do not want to wait a long period of time for video buffering. So the challenge is to deliver the video to the user with minimum buffering time.

Next is efficient video data structure. Ideally, users should be able to access the video content quickly and easily. They do not really pay attention to where the video is stored and how it is delivered to them. They simply want to access interesting short segments of video. They often do not want to access the whole original video that contains a particular interesting clip. Therefore, the challenge is how to deliver short clips of video regardless of the complex video data structure.

Regarding flexible disk location, usually the video is stored in the local sever disc because it is fast and inexpensive. However, with the increasing number of videos, we cannot just put all of them into the local storage. Moreover, this would reduce the performance of the video server. So the question is: how do we provide flexible storage in terms of location and sizes?

Last is the efficient database schema. On the one hand, Circus Oz video is an archive video collection whose existing data may unstructured and incomplete. On the other hand, the system is expected to retrieve the existing Circus Oz data as much as possible. In addition, the database schema development should also be concerned with delivering the user's expected data, and designing the front-end interface of this system so that users can conveniently browse and search the video content. Therefore, the challenge is how to design an efficient database schema that can accommodate both existing data and user's expected data.

This chapter starts with descriptive statistics of the Circus Oz video collection that has been uploaded to the Circus Oz Living Archive site[1]. And then, in order to answer the aforementioned challenges, we propose an architecture for the Circus Oz video retrieval system and its components including: video server, application server, database schema, search function, and video processing.

3.1 Circus Oz video collection

Circus Oz has made hundreds of recordings of their performances, rehearsals, promotions, and TV appearances from 1975 to the present in many countries including Australia, New Zealand, USA, Mexico, UK, Italy, Brazil, Israel, Germany, Holland, Hong Kong, and Singapore. Most of these recordings have been uploaded to the Circus Oz Living Archive site and Circus Oz continue to load new videos to the archive.

On 24/07/2014 when RMIT handed over the Circus Oz Living Archive system to Circus Oz, the entire video collection archive comprised 1,074 videos totaling over 1,000 hours of viewing. There are several types of Circus Oz videos in the collection including performances,

[1] http://archive.circusoz.com

promos, rehearsals, TV appearances, TV commercials, and other video types. Statistics of video types of the whole Circus Oz video collection can be seen in Figure 4.1. Most videos in the Circus Oz collection are performance videos: that is, 707 videos or 66% of the total number of videos. The promos, TV appearances and TV commercial videos have been made for advertising purposes. Other video types including documentaries and backstage videos contain non-show recordings associated with travelling, preparing the equipment, and setting up the stage.

Here we focus only on performance videos as these comprise the majority of videos in the collection and will be used for experimental purposes. The distribution of Circus Oz performance videos by year is shown in Figure 3.2. As can be seen, at least one video is released every year and, on average, there are about 19 performance videos per year. Starting from 1993, Circus Oz has produced at least ten videos per year and the highest numbers of videos uploaded were from 2006 and 2012, being 58 and 68 videos respectively.

Typically, the Circus Oz performance videos are lengthy, lasting anywhere from less than half an hour up to three hours. Figure 3.3 shows the Circus Oz performance video statistics according to their duration. The length of 322 videos is between 1.5 hours and 2 hours. Several videos are less than 30 minutes duration; in most cases, these are incomplete video shows which might be only the first half of the show. Also, several videos are more than 2.5 hours long, and these lengthier videos most likely include pre- and post-show recordings.

The Circus Oz video collection is accessible to two main categories of users: public users and internal users. A public user is any general user who visits the website, while an internal user is a user who is associated with the Circus Oz company including: the Circus Oz community, Circus Oz employees, and administrators. All users can access the public-access videos while non-public-access videos are restricted to certain users on certain conditions.

As of 24/07/2014, Circus Oz has published 80 videos accessible to public users. These public videos contain different types of recorded videos as shown in Figure 3.4. The publicly available performance videos range from 1979 to 2008 as shown in Figure 3.5. Circus Oz has released at least one video per year (up until 2008) of their whole collection for public access.

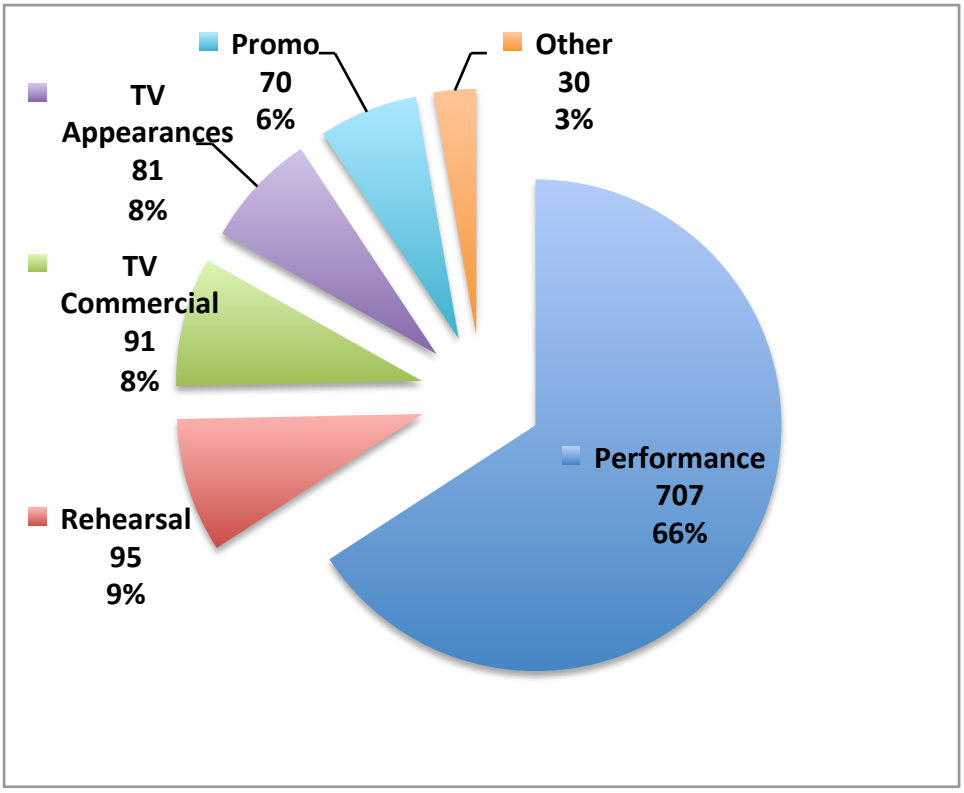


Figure 3.1: Types of videos in the Circus Oz video collection

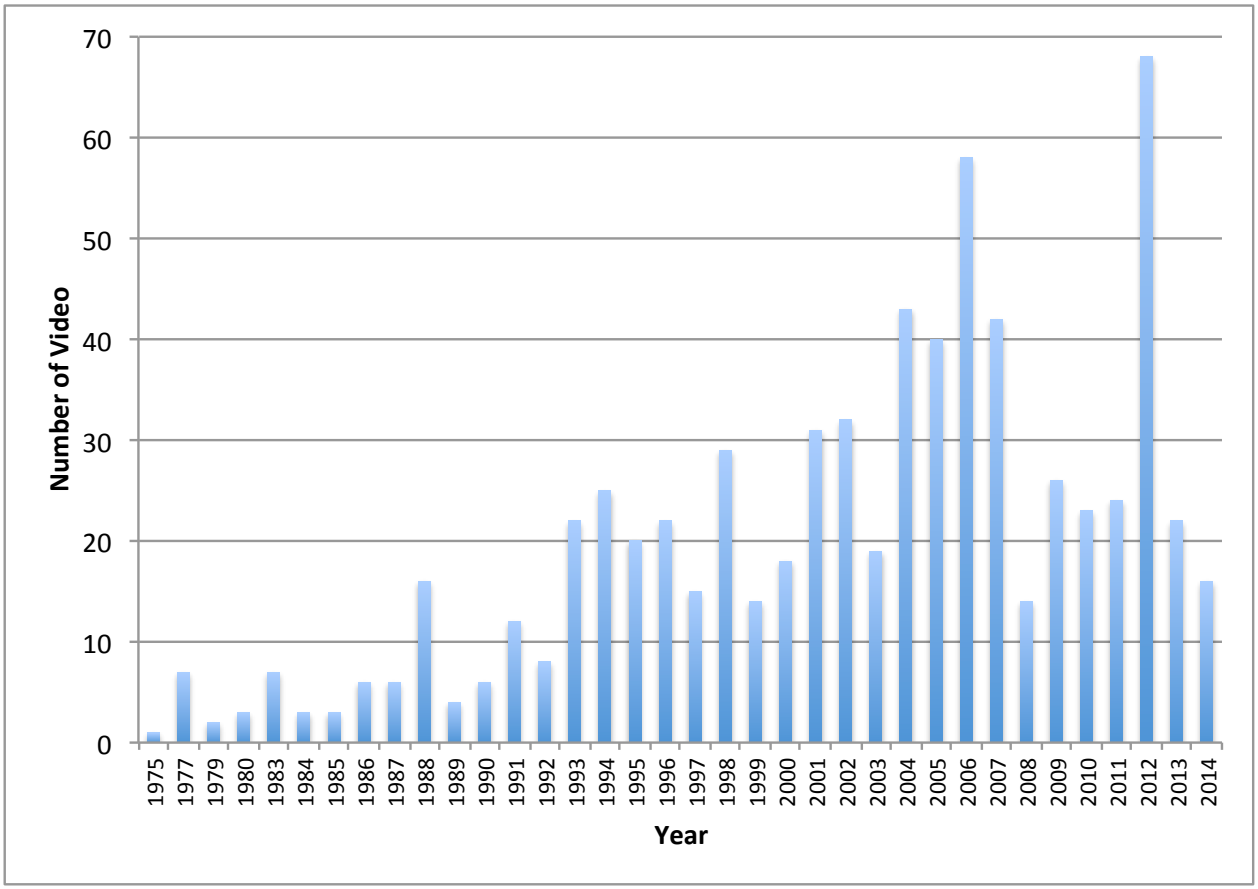


Figure 3.2: Distribution of the Circus Oz performance video collection by year

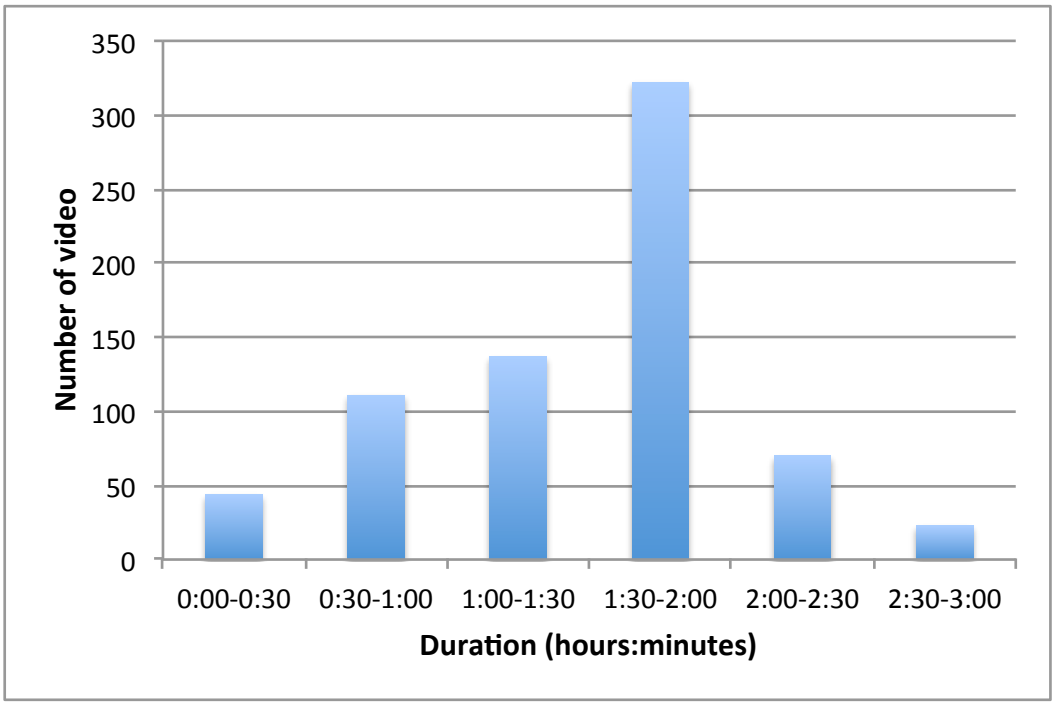


Figure 3.3: Circus Oz performance video type by duration (hour)

3.2 System architecture

System architecture is the general view of the system configuration and its components. There are three main components of the Circus Oz video retrieval system as shown in Figure 3.6. They are: video server, Circus Oz application server and end-user. The video server that manages all videos consists of three components: video application, database, and video storage. The Circus Oz application consists of a front-end application and a back-end application. The front-end application is a web interface enabling the end user to interact with Circus Oz application while the back-end application contains video processing features. The video application interacts with the video server through the video server API; the end-user requests and retrieves video to the front-end application server through the user's browser.

The video server is the main component of this system architecture. It stores and manages the entire video collection. This video server is not exposed directly to the users. Hence, it is secured from public user access. As the video server is dedicated only for the video management server, the performance of this server is not affected by processing on other

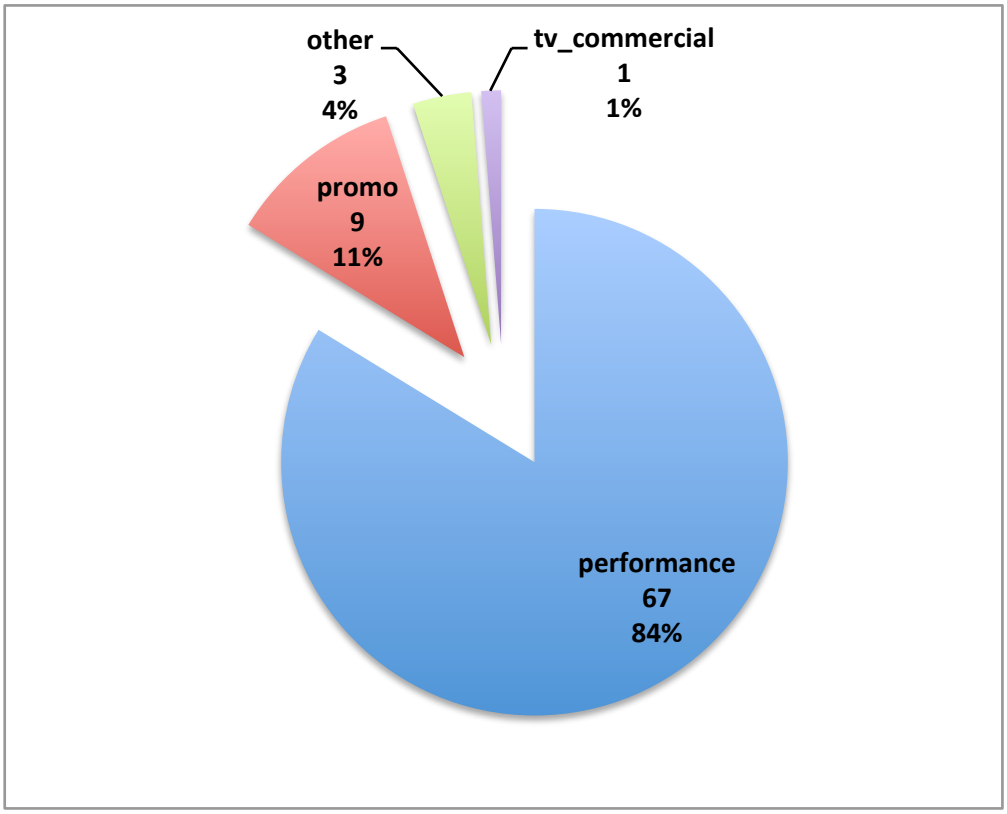


Figure 3.4: Circus Oz public video statistic by video type

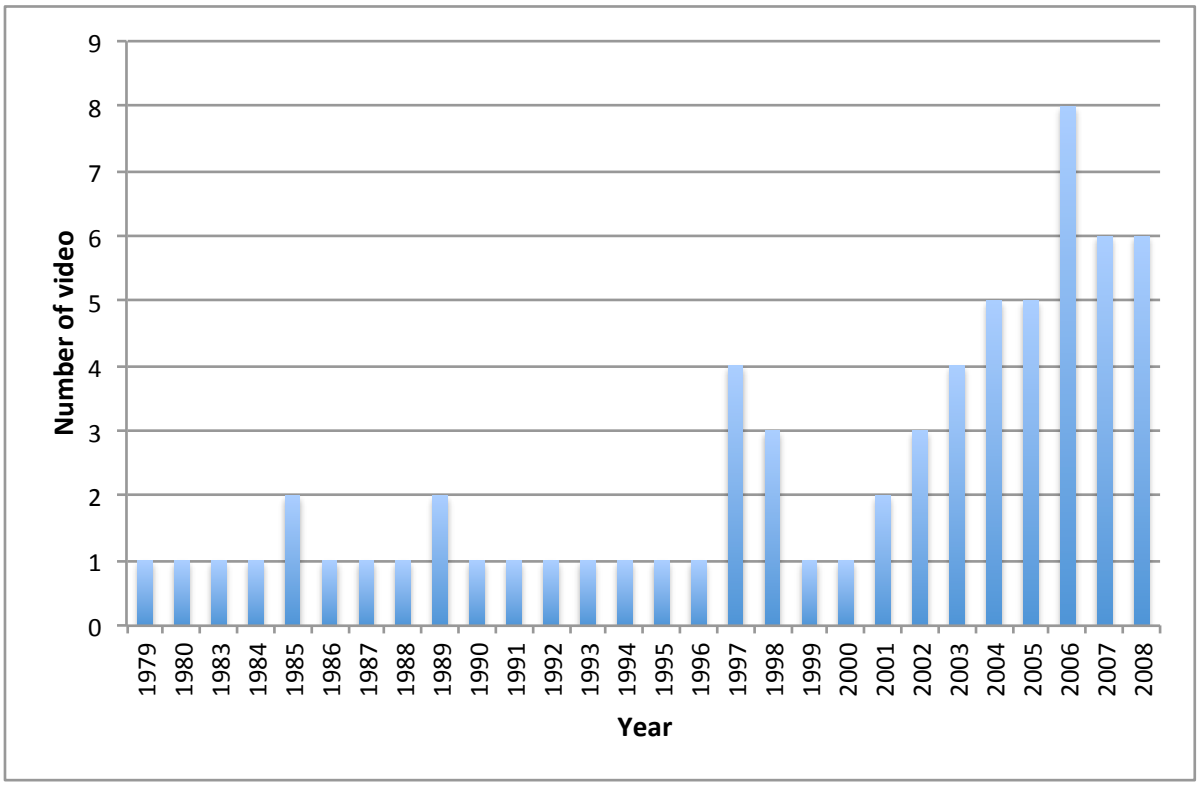


Figure 3.5: Circus Oz public video statistics by year

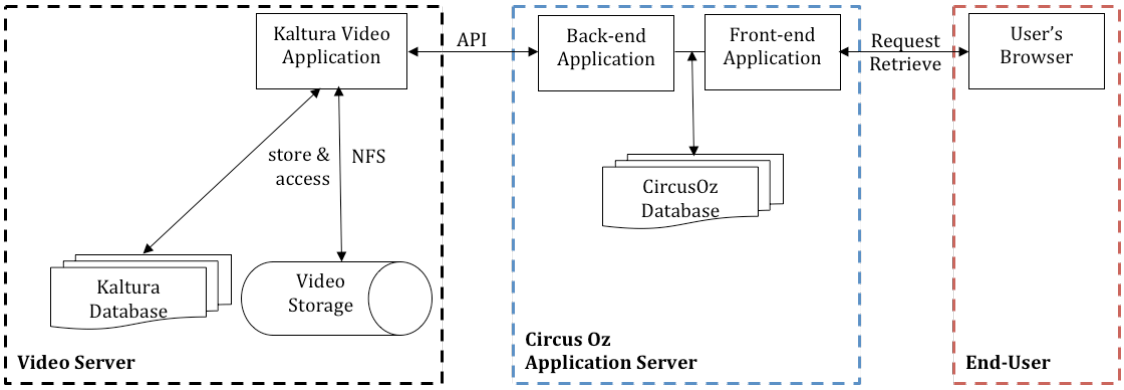


Figure 3.6: System architecture

servers: web server and database server. Furthermore, the video file is not stored on the video server but it is saved on a dedicated storage server. Therefore, its storage size is flexible so that it can adjust without any interruption to the video server. The storage server is a network drive for storing all Circus Oz video files and it is attached to the video server.

The application server is another main component of this system architecture. This server hosts the web-based Circus Oz application enabling public users to interact with this application through its user interface. The Circus Oz database is also stored in this server containing application data such as users, videos, clips, and comments. The Circus Oz database and its web-based application are installed in one server to allow fast loading and storing of data related to the web-based application. Furthermore, the Circus Oz search engine is also installed on this application server to speed up the search of the Circus Oz database in response to user queries.

With this system architecture that splits the video server and application server, both servers can run independently and be transparent to the user. If we want to change the configuration of one of the components of the system architecture, we do not need to re-configure the whole system. Furthermore, one of the components of the system can be upgraded without affecting the other components of the system. This system architecture also hides the video server from the users as users can interact only with the application server. That means that users would not realize that the video server is behind the application server. Hence, we can upgrade or completely change the video system without affecting user interaction with the whole system.

3.3 Video server

The main function of the video server is to manage and deliver the requested video to the users. In the Circus Oz video retrieval system, the video server delivers the video content to the user as requested by the application server. On the one hand, the video server requires video management capabilities such as: upload video, download video, and edit video meta-data. On the other hand, the video server requires the video processing capabilities such as video transcoding and image thumbnail generation. Furthermore, the video server needs to

handle video storage and effectively deliver video to the user.

The video storage is also a matter of concern. Video files can be stored on a local disc or stored on the network volume of a dedicated video storage server. Storing video files on a local disc may lead to a reduction in the video server performance. This is particularly the case if many users are accessing the server while the server is simultaneously delivering video to them. Hence, a dedicated video storage server is more suitable for a large video collection.

The problem is: how do we select a suitable video server for the large Circus Oz video collection? One requirement is that the video server should be able to offer secure access. The video server also has to be fully controlled by the admin user. This means that the admin user can manage which videos can be accessed and which cannot be accessed by specific user groups. The video server should also be capable of handling multiple video formats and resolutions.

According to the above video server requirements, we undertook some research before deciding which video server application to use for the Circus Oz video retrieval system. Unfortunately, the development of a video server was not possible as it was outside the scope of this research. An alternative solution would be a hosted video on a public video server such as Youtube[2] and Vimeo[3]. However, there are copyright issues associated with the audio and video contents of the Circus Oz collection. Circus Oz wants to keep some of the videos as private access for some users for certain periods of time. Moreover, specific features are needed in the Circus Oz video retrieval system such as custom video data fields and video segmentation. Hence, the public video server is also not suitable for this system.

Another solution would be to have an open source video server platform that can be freely customized and installed on the server. We use the Kaltura community edition video platform as a video server for the Circus Oz video collection. Kaltura[4] is an open source video platform. We installed and configured the Kaltura video platform on our server, so that, this video server has the following capabilities.

First is the secure video server. The open source video server can be downloaded and

[2]http://www.youtube.com
[3]http://www.vimeo.com
[4]http://www.kaltura.org/

installed as a dedicated server. This feature is quite important for a video archive system such as that of Circus Oz as we can have full control of all video content, operating systems and all related software. We can also determine which videos are available for public users and which videos are accessible to specific users. In addition, it is also easier to maintain the video collection on a local server. Uploading a lot of video is faster when we upload the videos from the local network where the server is located.

Regarding video access control, the video server has a video access feature that controls who can access a video at a specific time. Hence, we can allow all users to access a particular video and we can also prevent users from watching that video. Moreover, the expiry time of access can also be set for each video, enabling us to configure which video can be accessed at certain dates and times. In the Circus Oz video collection, public video access is usually given for videos that do not have copyright issues while, private video access is given to the videos that still have copyright issues. The copyright issues are associated with performers and music. Once Circus Oz resolves the copyright issues, we can then allow public video access. In fact, the new uploaded video is quite often published as a private video access type because its content needs to be checked first.

Next is the flexibility of video storage. The video server can also manage the video location storage. By default, all videos are stored in the local disc of the video server. As Circus Oz has a large video collection, a substantial amount of disk space is required. The large video file can be saved on an external or network disk location. Technically, the video files can be saved anywhere as long as the video server has a network access to the files. In this system, we set up the video storage server and it is attached to the video server through a network file system (NFS) protocol.

Last is disk usage efficiency. In order to reduce the disk space, each video will have only one entry in the video server. Although each video is segmented into meaningful short clips, one entry per video in the server is still applied. The reason is that the video is actually not physically segmented. Instead, the logical segmentation is done by entering the start time and end time of a particular clip in the database. After that, we use that data to load the video jump to a specific start time. Hence, the user can still browse the segment of video

but the video will still be stored as one. This feature was created by transcoding the video into MP4 format with h264 codec, enabling fast loading and fast seeking. The fast seeking feature helps to find the video at any position without the need to buffer the whole video up until that point.

The details of the implementation of the video server are presented in Appendix B.

3.4 Circus Oz application

The Circus Oz application is one of the main components of the whole Circus Oz video retrieval system. It is integrated with other systems such as the video server, video storage, and database systems. They collaborate to serve the users as shown in the system architecture diagram (Figure 3.6). This is how the system works. First, a user interacts with the Circus Oz application through the browser application by submitting a search query for example, then the Circus Oz application processes the user's query. Based on that query, the Circus Oz application contacts the database server to obtain the video metadata and at the same time, it also contacts the video server to obtain the video. Finally, the video is delivered to the video player on the user's browser application.

Specifically, the video server and the application server are connected through the API provided by the video server. For example, on loading a video content, the application server contacts the video server because the application server does not have the video files because the video_id data is stored in its database. The application server can retrieve the metadata from its database and send it directly to the user. Once the user requests the video, the application server searches for the video_id of that video and contacts the video server through their API and requests the video content. After that, the video server delivers a URL of that video to the application server. This URL contains the video location on the video storage server. In the application server, the video will be loaded into a video player where that URL is a file source. Finally, the video player will load the video directly from the video storage server.

As mentioned in the previous section, the start time and the end time of the clips are stored in the database. Once the user requests a particular clip, the Circus Oz application

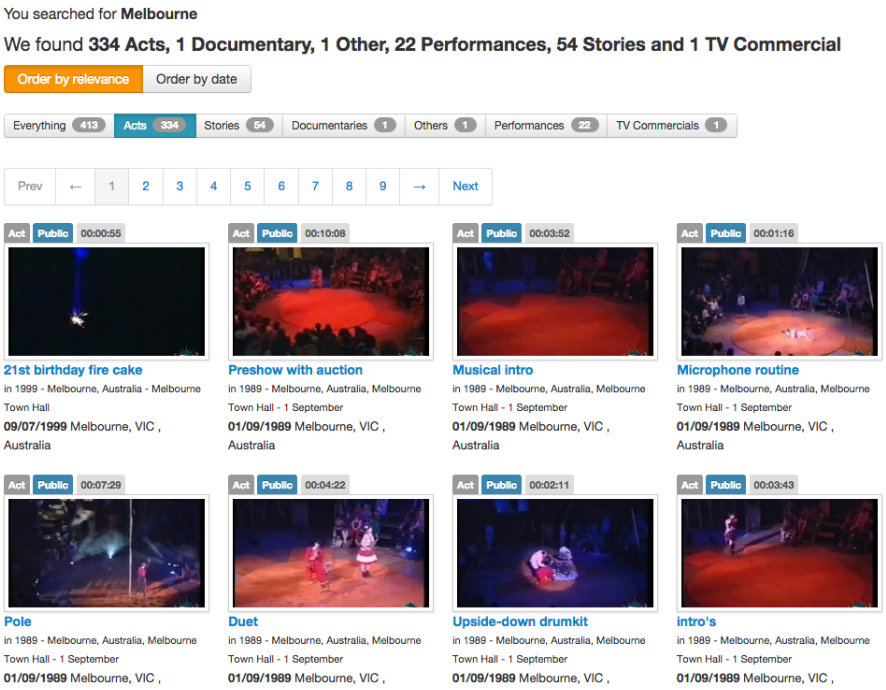


Figure 3.7: A search result in different views

will load the start time and end time of that clip. After that, on the video player interface, the video will be loaded and jumps to that start time of the clip. This was made possible because all videos were transcoded into MP4 format with fast loading and fast seeking capabilities. In the Circus Oz application, we defined the clips as a short segment of video. This short segment is not only video content (acts) but also includes user comments, tags, and performers.

The details of Circus Oz video application specifications and its interfaces are described in Apendix C.

The benefit of this video application is that a simple database can be represented in different ways. A performance video contains only one single video file on the video server. However, we display that video file in different ways including acts, stories, and performances views (Figure 3.7). Furthermore, when a user submits a search query to the application, the system will match the query against the data in the video table, the act table, the performers table and the story table on the Circus Oz database. Therefore, the search result can be displayed in different views: videos, clips, and stories.

3.5 Circus Oz database

A suitable database schema is needed for the implementation of the logical video segmentation and is mentioned in the video server feature. The logical segmentation video stores the segmentation points in the database instead of segmenting video files physically. The proposed database schema has to be compatible with the existing data structure so that the existing data can be imported into the database. Also, the proposed video-clip relation needs to be included in the database schema.

3.5.1 Existing data

The existing Circus Oz data includes show reports, human resource personnel and catalogue data. These come from several sources with different formats including FileMaker database, Spreadsheets, and text documents. The details of existing Circus Oz data are given in Figure 3.8.

First, a show report is the data of the show in spreadsheet document form. It mainly reports the show including who is in the show, the order of acts, location, etc. This is very valuable data and can be used in the new database system. This show report is filled out on computer in conjunction with the Artistic Director.

Next is the Circus Oz human resource personnel data, which is structured data in FileMaker database format. One main table (season table) and three other tables that relate to the performers (show, personnel, and personnel data tables) are in this database.

Other existing data are a video catalogue and a tape description. They are on spreadsheets and text documents. These documents were created mainly when Circus Oz was converting video from analog to digital video format. During the conversion process, the analog videotapes were labeled and the process was recorded in the catalogue file.

Based on the existing data, table relations can be created as shown in Figure 3.8. The solid lines indicate that there is a data relation between two or more tables. The broken lines indicate that there is no relation but the contents in both tables are actually related. As can be seen from the figure, the database relation taken from the personnel data is shown clearly. The primary keys are session_id and person_id. In addition, although video catalogue and

tape desc is not in the form of a database, there is a potential key connection: tape_id.

There is no connection between the video catalogue and the season table. However, we can create a connection based on the date field in both tables. The connection is that if the date in the video catalogue is within the range of date from and date to in the session table, we can then populate the session_id in the catalogue table. As a result, we can generate the performers who appear in the video.

The remaining issue on that database relation is the show report. Although the Show_No and the Tape_id are the same thing, they have different formats. Hence, we have a list of shows in the video collection.

3.5.2 Video-Clip relation

Here we introduce a concept of video-clip relation to support Circus Oz database development. A video is a whole video of a circus show while clips are any short segments of that video. The idea here is that we convert everything to a clip. A clip is the main object in this concept although clips are part of a whole video. Therefore, we expose the video through the clips. Any other object related to the segment of video is also a clip and can include:

- Video: The video itself is a clip, which has duration from the begining to the end of the video.
- Act: Act is part of the circus video show.
- Comments: relating to a part of video acts or the whole video.
- System-generated clip: black frames, applause, and mute sound clips

As we can see from Figure 3.9, only the clip object is exposed to the user. For example, if a user requests to view clip_id 2, which is a group balancing act, the system will load the video of that act which is video_id 1. Then the application server will request that video from the video server through its Kaltura_id (0_l5qn5myi). Finally, video server will deliver video_id 1 to the user. On the Circus Oz application, the video player will look up the start time and end time of that clip. The video will move forward directly to the start time of

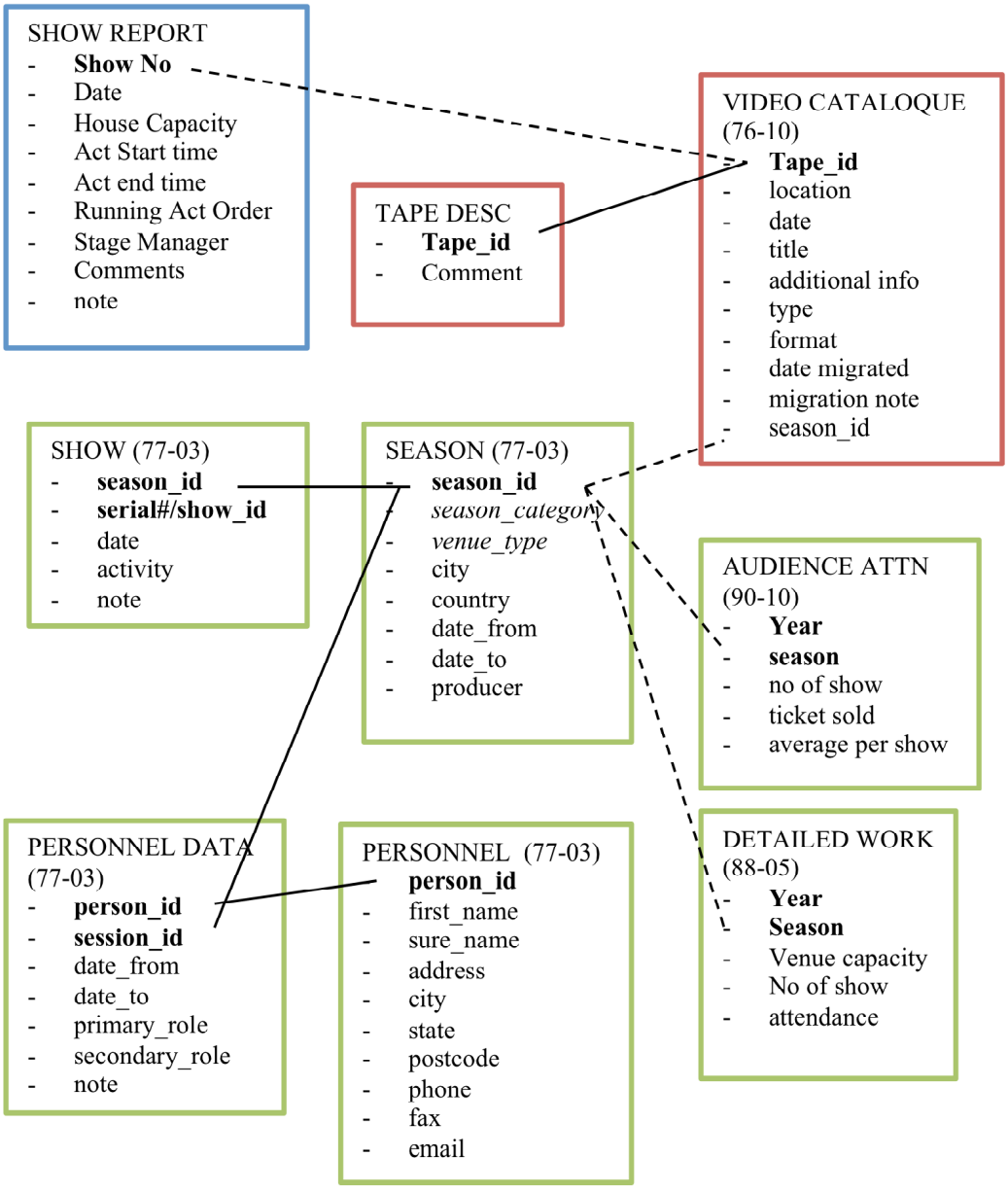


Figure 3.8: Existing Circus Oz data source

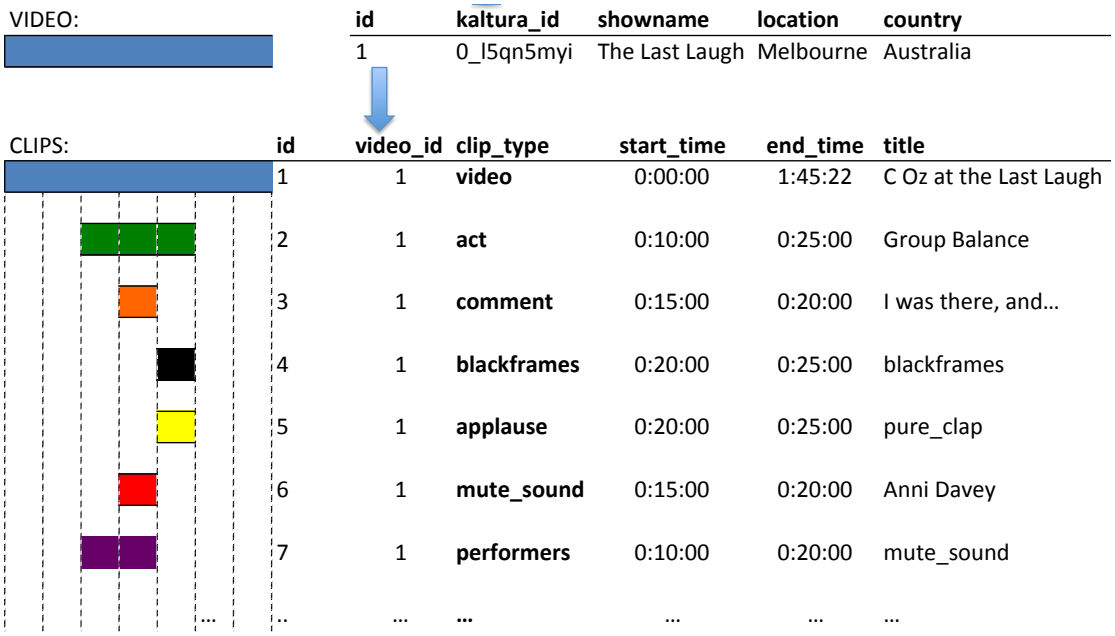


Figure 3.9: Example video – clips relation in Circus Oz database

the clip instead of starting from the beginning of the video. Moreover, the application server will also look up any clips that are related to that clip. That is, the clips that have the same video_id and are within the range of the start time and the end time of the clip which is video_id 1, start time=0:10:00 and end time=0:25:00. The clips that match these criteria are:

- clip_id 3: comment
- clip_id 4: black frames
- clip_id 5: applause
- clip_id 6: mute sound
- clip_id 7: performers

Therefore, the system will return all information related to that video including comment, black frames clip, applause clip, mute sound clip, and performers data.

On the interface it will appear differently depending on the purpose of the clip. For example, black frames and applause are system-generated clips. These clips will not be

shown on main video player page for public users. However, these data are useful for admin users to segment the video into acts. The other clips which are comments, mute sound and performers will be shown to the user. That is, the content of the commentary on the clip. The audio will be muted once the player hits mute sound frames and the performers who perform on that video also will be shown.

3.5.3 Database schema

For the prototype, we designed the database based on the video and clip structure. A video is the main table while a clip is a secondary table. In the main table, a video is always recorded in one record. However, in the secondary table, this video can have multiple clips including acts, comments, images, etc. This video-clip structure is quite flexible and therefore able to accommodate any additional clip type in the future.

Based on several initial database prototypes (see Apendix C), we designed the final database schema as shown in Figure 3.10. Each video has one entry in the video and the Kaltura tables. The video table stores the description of the performances; the Kaltura table stores the metadata of the video file. The relation between these two tables is a one-to-one relation. Therefore, one video can have only one file in the Kaltura video server.

The video table is also the parent table for the clip table. That is, each video can have a number of short video segments in the clip table. The short segment of a video could be the data about performers or acts. The performers'data is stored in a personnel table and its details are stored in the people data table. The act clip is stored in the act data table and its category and type are stored in the act table.

In addition, we introduce the term *collection*. A collection is a list of clips that is created by the user. This can be any clip in a video and does not always come from one video. It could come from different videos. Moreover, that clip can belong to a user. So, in the clip table there is a user_id. The collection table has a direct relationship with the clip table through the clip_id. One collection contains the id of many clips, and also belongs to a user identified by the user_id in collection table. To both the collection and the clip tables, we add a tag field. This enables the user to tag a clip or collection, making it easier to search

than if it were not tagged.

The final database schema of this prototype is shown in Figure 3.10. In this relation concept, the relationship between video and act is simplified as a video and clip relation. The clip is not only an act, but can be any data associated with the video including: comments, performers, tags, images, etc. Therefore, in the future we can add other new related video data without changing the database schema. An example of the video-clips relation in Circus Oz database is shown in Figure 3.9.

3.6 Search functionality

Circus Oz has a large collection of videos and metadata. Browsing the video database using the provided metadata (date, title, and location), may not be preferred by some users as it takes some time to obtain the actual video clip. Instead, users may prefer to type the query based on their knowledge about Circus Oz and obtain search results enabling them to choose which video to watch. However, given the huge video collection and complicated data structure, it is a challenge to develop the right search function. Here, we describe the Circus Oz search functionality including the search engine and main key fields indexing.

In order to resolve the above issues, we used the Sphinx search engine as part of the Circus Oz application. Sphinx[5] is an open source search server. The details of the search implementation are presented in Appendix E. In order to implement the Sphinx search server, the search index and field weight need to be considered.

The search index are the data or fields in the database that are required for search function. These data are usually the main key fields that are of most interest to the users. For the Circus Oz database, users are most likely to be interested in video data including: video title, description, tags, video type, country, city, venue, and performers. In addition, other key fields such as id, clip_id, and tape_id, also need to be indexed as admin users are usually interested in those key fields.

The field weight needs to be established since some fields are more important than other fields. The most important field has more weight than less important fields. For example,

[5]http: //sphinxsearch.com

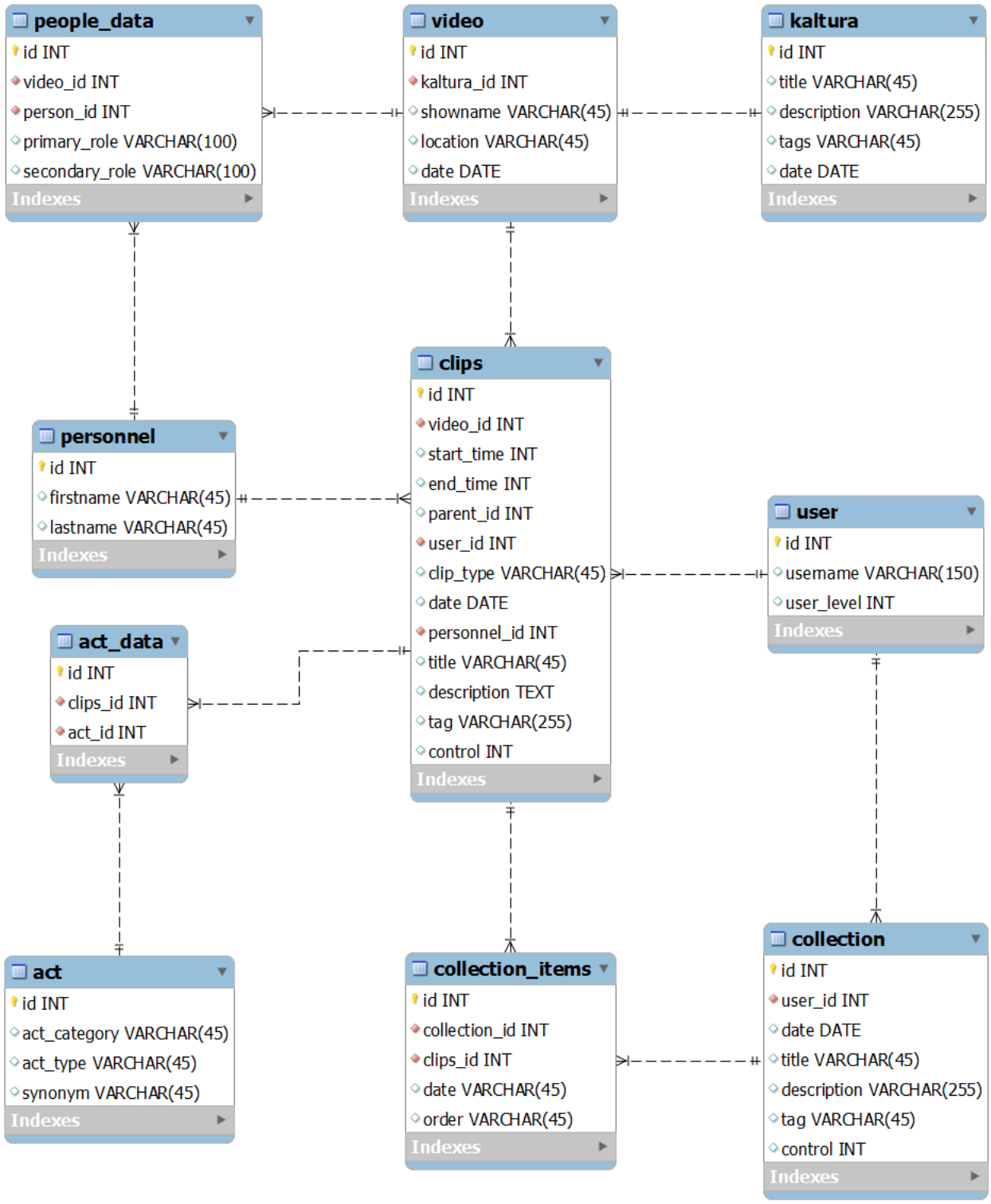


Figure 3.10: Final Circus Oz database schema

No	Field Name	Weight
1	Title	5
2	Person	5
3	Video_id	5
4	Admin Tag	4
5	Date	4
6	City	4
7	Country	4
8	Venue Type	4
9	Venue Name	4
10	User Tag	3
11	Description	3
12	Username	3
13	Video Type	3
14	Clip Type	3

Table 3.1: The field weight on Sphinx Search

the video's title has more weight than the video type as the title is unique to each video, while many videos could have the same video type. The field weight table for the Circus Oz video database is shown in Table 3.1. To decide which field and their weight to be set on the search indexing, we divided the search index into three groups with different weights: 5, 4, and 3. These weights are heuristically determined by trial and error with a more important field having a higher weight than a less important field. For example, the title, person, and video id are the most important as they are quite unique and used by admin user to tidy up the database.

3.7 Video processing

In the Circus Oz video application, the videos need pre-processing before they are uploaded to the video server. The main reason for this is to protect the video content of Circus Oz including both music and copyright. The background music in some Circus Oz videos may be restricted for public users; therefore, a Circus Oz logo needs to be put on each video to prevent violations of copyright. In addition, video pre-processing is needed to merge the split video into one video. In fact, several Circus Oz performances were recorded on two (or more)

Video Pixel Wdith Size	Logo Pixel Width Size
320	30
480	40
640	50
720	60
1280	70

Table 3.2: Watermark Circus Oz logo size guidelines

Figure 3.11: Watermark Circus Oz logo

videotapes. The video pre-processing tasks include: video watermarking, muting of sound, and video merging.

3.7.1 Video watermark

A video watermark is required in order to avoid copyright issues associated with the Circus Oz video content. A video watermark is a Circus Oz logo that is placed at the bottom right hand corner of the Circus Oz video. We use the FFMPEG video platform to do this. The size of the Circus Oz logo on video varies depending on the video size. Table 3.2 provides guidelines regarding logo size; Figure 3.11 shows the different sizes of the Circus Oz logo.

3.7.2 Sound muting

In order to avoid copyright issues regarding music, the background music in a particular video needs to be muted. An administrator can select the segment of video to be muted as shown in Figure 3.12. After that, the system will run a script in the background to mute the video. The original video will be kept as is and the mute video will be added to video server. Therefore, there will be two versions of that video. One is the original video that can be accessed by only the administrator and the other one is the mute video version that can

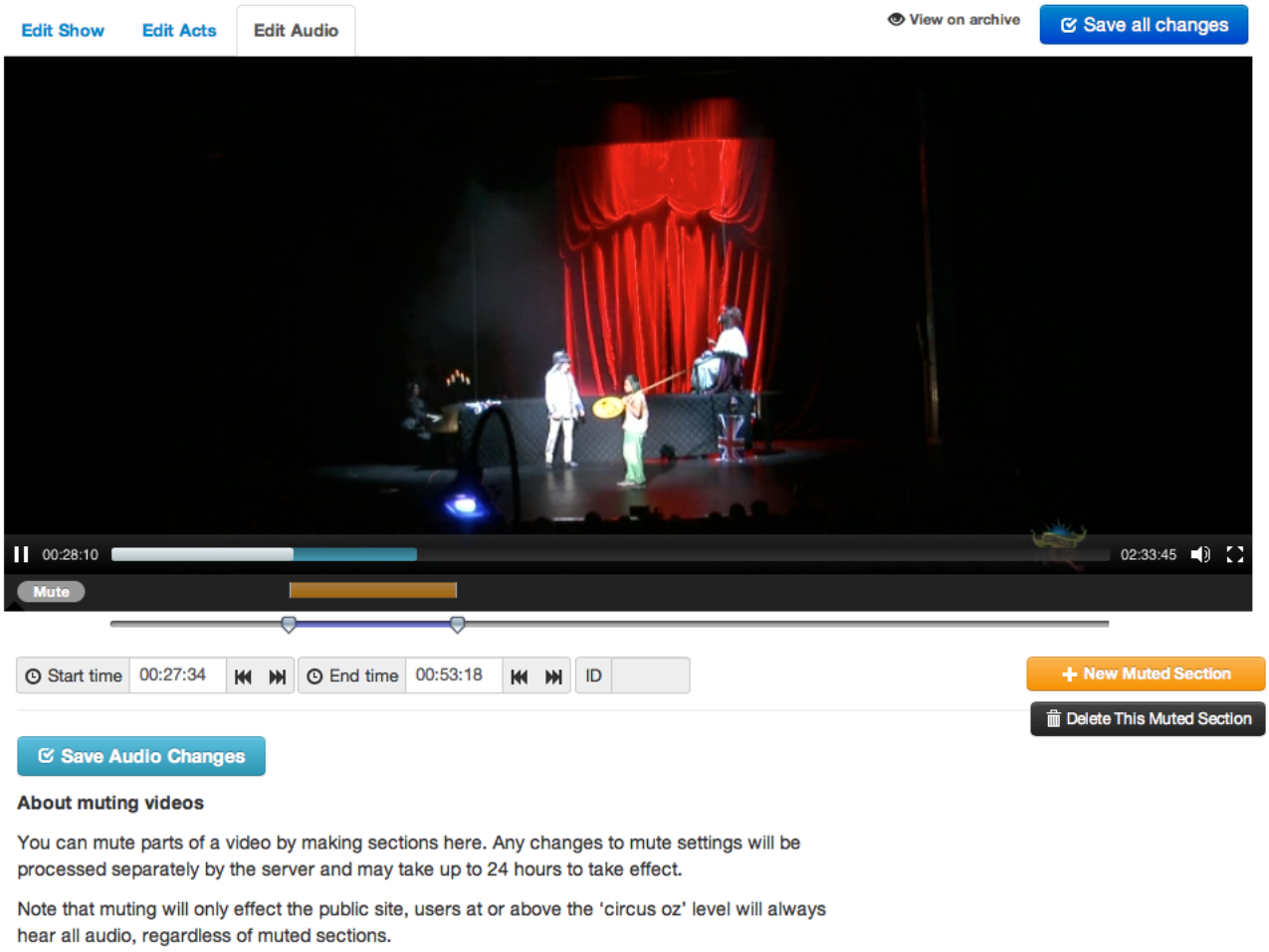


Figure 3.12: Interface for muting video on Circus Oz application

be accessed by all users including public users.

3.7.3 Video merging

The video merging process is also required if the recorded videos are split across more than one video file. The process is as follows: each video is first converted into "mpg" format before being merged into one file. The reason is that we cannot merge two "mp4" files into one. After that, all "mpg" files are merged into one "mpg" file. Finally, the merged "'mpg" file is converted back into an "mp4" file.

3.8 Summary

In this chapter, we have outlined the Circus Oz video collection and its video system including its six components that address the challenges associated with the development of a video retrieval system. These components are discussed below.

Circus Oz video collection. The Circus Oz video collection comprises 1,074 videos totaling over 1,000 hours of viewing. Most of them are the performance type video (66%). The duration of most Circus Oz performance videos is between 1.5 and 2 hours.

System Architecture. The Circus Oz video retrieval system comprises the video server and the Circus Oz application. Both servers can run independently and are transparent to the users.

Video server. Four main features to consider when selecting or developing a video server have been explored. They are: secure access to selected videos, independent from application server, flexible storage location, and disk space efficiency.

Circus Oz application. The Circus Oz front-end application server is a gateway between the content video server and the user as it delivers video and data through its interface. The simple database can be represented in different ways on the front-end interface.

Circus Oz Database. When designing the Circus Oz database schema, existing Circus Oz data and proposed video-clip relation were considered. The final database relation concept is a video and clips relation. The video is recorded one entry per video, while clips could be multiple entries. The clip consists not only of video content (acts) but also can include other segment metadata such as comments, tags, images, and performers.

Search functionality. A search function has been developed enabling users to explore the Circus Oz video collection conveniently. The search field index has been selected and weighted based on the importance of the field to the users.

Video processing. Circus Oz videos need to be pre-processed before they are uploaded to a video server to protect the video content. They are: video watermark and sound muting to deal with copyright issues; and video merging to merge split video footage.

Chapter 4

Circus Oz Dataset

Several different applause sound datasets have been used to evaluate applause sound detection technique performances. These datasets include: an auditory training system [Lesser and Ellis, 2005], meeting speech [Manoj et al., 2011; Li et al., 2009], and music concerts [Sarala et al., 2012; Marmaroli et al., 2013]. Unfortunately, none of them has been published and made available for experimental purposes. We set up the Circus Oz applause sound dataset to evaluate the performance of the proposed applause detection technique described in Chapter 5.

Setting up an applause dataset for Circus Oz videos is challenging. The reason is that the typical Circus Oz performance videos are lengthy and contain quite a lot of applause on one video; moreover, it is often difficult to find when and where the applause occurs. The applause sound could occur simultaneously with other sounds which could be mistaken for applause. In addition, finding the exact start time and end time of the applause sound contributes to the difficulty of this task. Furthermore, the Circus Oz applause dataset is published online[1], enabling other researchers to explore and test the performance of applause detection techniques.

We also develop a dataset to evaluate the performance of the proposed video segmentation technique described in Chapter 6. The purpose of the video segmentation technique is to segment a circus video into acts. This dataset also taken from the Circus Oz video collec-

[1] http://www.cs.rmit.edu.au/~jat/circusoz/

tion includes single applause type, image comparison, black frame, and end-of-act datasets. However, this dataset has not been published online, as some of the Circus Oz videos in this dataset are not publicly available.

This chapter explains the development of an applause sound dataset and video segmentation datasets. It starts with the development of the dataset with multiple applause types, followed by the development of the video segmentation dataset. Finally, a summary is provided of the datasets developed from the Circus Oz video collection.

4.1 Multiple applause type dataset

4.1.1 Video dataset

This applause sound dataset is made up from videos that are accessible to the public user. So anyone can access the videos in this dataset. Of the 67 performance videos available to public users, we have selected 12 performance videos for our dataset. We consider only the performance video type because this video type contains more applause sound than other video types. We took only one video from each odd numbered year from 1983 to 2005. They are: 1983, 1985, 1987, 1989, 1991, 1993, 1995, 1997, 1999, 2001, 2003, and 2005. The durations of the 12 videos in this dataset range from 1.5 to 2 hours. The total duration of the whole dataset is 21 hours, 19 minutes and 53 seconds of viewing. Furthermore, we divided the dataset into 2 subsets: a development and a test dataset, so that each subset has 6 videos. The development dataset is used to fine-tune the parameters established for the proposed method; the test dataset is used to evaluate the performance of the proposed method.

The complete list of videos in this dataset is displayed in Table 4.1 and shows the Audio ID, Title and URL, Year, Duration, and Subset. The Audio ID is the dataset ID for each video in the format: XX_subset, where XX is an increment number from 01 to 12, while the subset is either for the development set or for the test set. The Title of the video includes name, location, and date of the circus performance. The URL is taken from the Circus Oz living archive site. The duration is the length of each video.

Audio ID	Title and URL	Year	Duration	Sub set
01_devl	Melbourne, Australia, Big Top, Princes Park - 27 February, http://archive.circusoz.com/clips/view/33	1983	1:47:36	devl
02_test	Alice Springs, Australia, Araluen Arts Centre - 24 July, http://archive.circusoz.com/clips/view/36	1985	1:45:12	test
03_devl	Sydney, Australia, Big Top, Seymour Centre - 9 January, http://archive.circusoz.com/clips/view/42	1987	2:02:17	devl
04_test	Melbourne, Australia, Melbourne Town Hall - 1 September, http://archive.circusoz.com/clips/view/45	1989	1:42:49	test
05_devl	Melbourne, Australia, Melbourne Town Hall - 14 September, http://archive.circusoz.com/clips/view/58	1991	1:40:07	devl
06_test	Melbourne, Australia, Flying Fruit Fly Circus Tent, City Square - 1 May, http://archive.circusoz.com/clips/view/79	1993	2:00:45	test
07_devl	Sydney, Australia - Flying Fruit Fly Circus Big Top, Moore Park, http://archive.circusoz.com/clips/view/107	1995	2:04:33	devl
08_test	Sydney, Australia - Big Top, Moore Park - 23 January, http://archive.circusoz.com/clips/view/134	1997	1:44:14	test
09_devl	Melbourne, Australia - Melbourne Town Hall, http://archive.circusoz.com/clips/view/147	1999	1:49:21	devl
10_test	New York, USA, New Victory Theater - 1 June, http://archive.circusoz.com/clips/view/6	2001	1:35:41	test
11_devl	Melbourne, Australia, Big Top, Birrarung Marr, http://archive.circusoz.com/clips/view/295	2003	1:33:31	devl
12_test	Melbourne, Australia, Birrarung Marr - 26 June, http://archive.circusoz.com/clips/view/248	2005	1:33:47	test

Table 4.1: List of video dataset

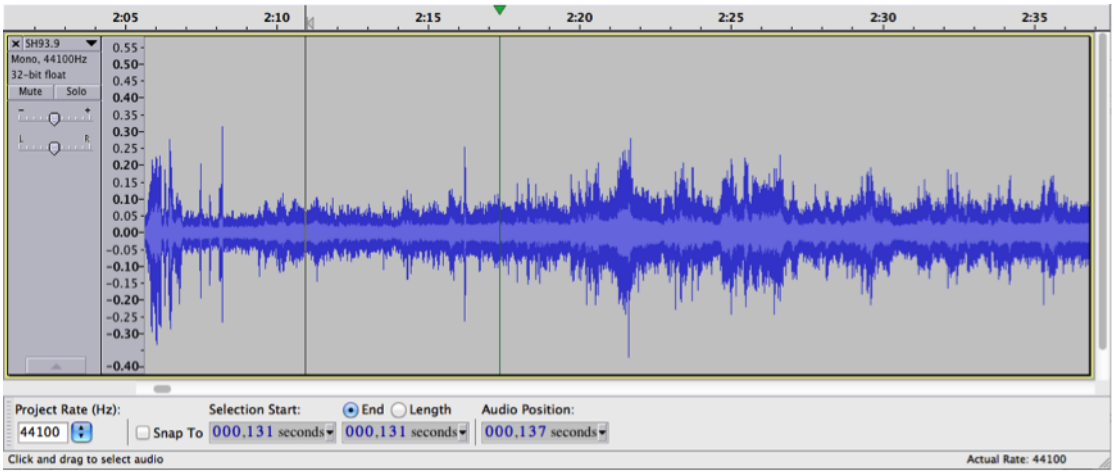


Figure 4.1: Wave graph on Audacity (R) recording and editing software

4.1.2 Ground truth

In order to verify the performance of our proposed applause detection algorithm, we also determine the ground truth of where and when the applause occurred in the dataset videos. We manually labeled the applause sound in each video. One person, the author of this thesis, was responsible for labeling the applause sound on each video. To find the applause sound, we listened to the audio content only, without watching the visual content. Instead, we focused on a wave graph provided by audio software to make the coding of the applause sound time more precise (Figure 4.1). The original audio quality varies from low to high quality. We use ffmpeg audio tools to convert the video file, mp4, into audio file, wav format. The converted audio profile is 44.100 Hz sample rate, mono audio, and 32 bits per sample.

The typical applause sound on a Circus Oz video is mixed with other sounds: music, cheering, laughter and speech. This is why we extended the applause class into two additional classes: 'less clap' and 'more clap'. The 'less clap' class is chosen if the other sounds are louder than the applause sound, while the 'more clap' class is chosen if the applause sound is louder than other sounds. The 'pure clap' class contains only the sound of applause. A single-person clap is excluded, as it is usually short and indistinct. Therefore, we defined four classes of applause sound on the Circus Oz videos: 'no clap', 'less clap', 'more clap' and 'pure clap' (as described in Table 4.2).

class name	description
no clap	The applause sound does not exist in this clip. It contains other types of sound instead, including: music, speech, cheering, silence, laughing, and mixing various types of sound.
less clap	This clip contains mixing between applause and other sounds. However, the applause sound is not dominant in the clip compared to other sounds. For example while music is playing, there is sound of applause in the background. Usually, this applause is not quite clear or the volume of this applause sound is low.
more clap	This clip contains mixing between applause and other sounds. However, the applause sound is dominant in the clip compared to other sounds. For example, while applause is occurring loudly, the music is still playing in the background.
pure clap	This clip mostly contains only applause sound. The applause sound could be low or high but there is little or no other sound in this clip.

Table 4.2: Applause class on Circus Oz video.

Given the nature of audio content in the Circus Oz live recordings, the four classes mentioned above can sometimes overlap with each other. One clip may contain more than one type of applause (no clap, less clap, more clap, and pure clap). We decided to label the class with the longest type of clap. For example, consider a 10-second clip that consists of 7 seconds of pure clap and 3 seconds of more clap; this clip would be labeled as pure clap because the pure clap duration is longer than the more clap.

The precision of timing of the start and end of labeled clips is in the number of seconds from the start of the recording. The reason is that it is quite difficult to listen and label the clip below one second. However, a second at the beginning and a second at the end of the clapping sound are quite often a combination of applause and other sounds including music, speech, cheers and silence. Hence, we decided to take the start time and the end time begining with the first second where the applause sound dominates than the other sounds.

Furthermore, it is sometimes the case that applause begins with one single person applauding, and the applause ends with one person applauding. As we exclude the single-person applause, the start time of the applause is not the single-person clap and also the end time of the applause is not the single-person applause.

After obtaining the initial ground truth from audio dataset, we re-check each applause clip resulting from the manual audio labeling described above. This is to make sure that the timing code and the label are both correct. Here are the steps:

1. Extract applause clips

The applause sound clips on each audio dataset are extracted using ffmpeg. This extracting process refers to the ground truth database consisting of start time, end time, video id, clap no and class of each clip. Each clip is then named in the following form: video id + clap no + class. For example: 026-001-more.wav meaning that this clip contains more clap class, taken from video id 026, clap number 1.

2. Re-listen to applause clips

Each applause sound clip is then listened to again two or three times and then compared with the audio original source.

3. Adjust the timing code

One of the criteria for re-checking the applause clip is the timing code: the start time and end time of the applause sound clip. At the beginning and at the end of applause, the sounds are listened to carefully. This is to ensure that both points contain the applause sound. Quite a few of the clips end with a single clap or no clap (music, silence, etc.) at the last second. Hence, the last second is removed. Some of the first seconds do not contain applause or contain a single clap. This first second is also removed. Some of the clips have a wrong timing code which is either too long or too short. In such cases, we check the original audio files and make the appropriate adjustment.

4. Re-label the class

Another criteria for re-checking the applause clip is the applause class. This is to ensure that all clips have been given the correct applause class. Each clip has to be labeled as one of three applause classes: less clap, more clap, and pure clap. Some clips are listened to again up to four times, particularly those with less clap and more clap. Sometimes it is difficult to distinguish between these two classes. The different audio quality of each video makes it more challenging to distinguish less clap from more clap task. The less clap or more clap clip between one video and another video could be slightly different. In addition, most pure clap clips contain the sounds of cheering or laughter. This type of clip is typically lengthy. As a result, it might contain other classes: less clap and more clap classes. If that is the case, we label this mixed clip as the most dominant or longer class.

4.1.3 Statistics

The statistics of the applause dataset are divided into two parts: the number of applauses and their duration. First is the statistics of the applause sound dataset based on the number of times that there is the sound of applause. Table 4.3 and Figure 4.2 show the percentage of whole applause sound dataset and the number of applause sounds respectively. As shown in Table 4.3, most applause classes in the Circus Oz video is less clap (44%) and more clap (40%) while the other class (pure clap) is only 16%. Similarly, statistics for the number of times there is applause in the video (Figure 4.2), show that less clap and more clap are the most frequent applause classes in each video, while the pure clap is in the minority. This is

class name	percentage(%)
less clap	44
more clap	40
pure clap	16

Table 4.3: Percentage applause class.

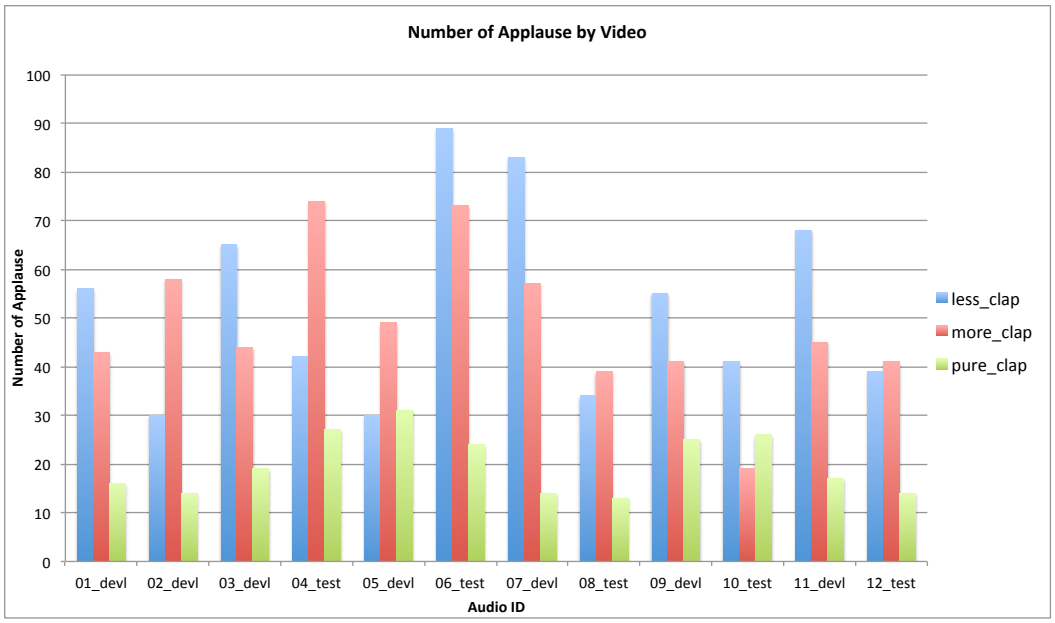


Figure 4.2: Number of applause by video

noticeable in dataset 07-devl where less clap and more clap classes are predominant with 80 and 60 applauses respectively, while pure clap is in the minority with only 10 applauses.

Now we consider the statistics of the applause sound dataset based on applause duration. Table 4.4 and Figure 4.3 show the percentage of whole applause sound dataset and applause sound duration respectively. As shown in Table 4.4, more clap is the longest applause duration in the whole applause dataset with 1 hour 25 minutes and 17 seconds long while pure clap has the shortest duration at 36 minutes and 49 seconds. Similarly, statistics of applause duration by video (Figure 4.3), indicate that more clap has the longest duration on every video except on video: 03 devl, 09 devl and 11 devl, while the pure clap has the shortest length in all videos except for the video 10 test.

class name	duration	percentage
less_clap	1:06:38	35%
more_clap	1:25:17	45%
pure_clap	0:36:49	20%

Table 4.4: Percentage duration applause class

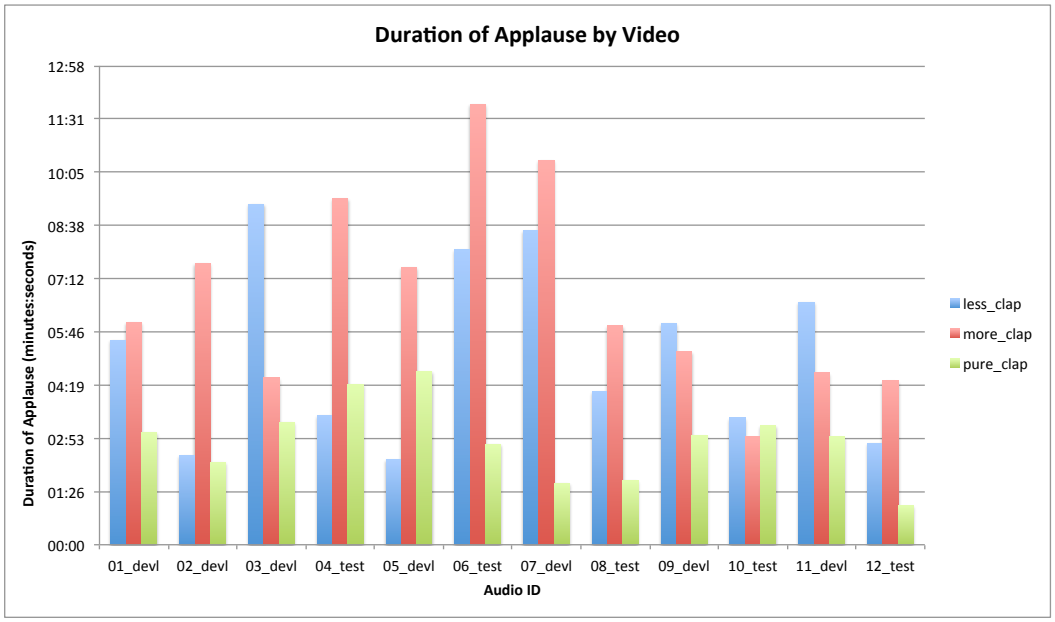


Figure 4.3: Duration of applause by video

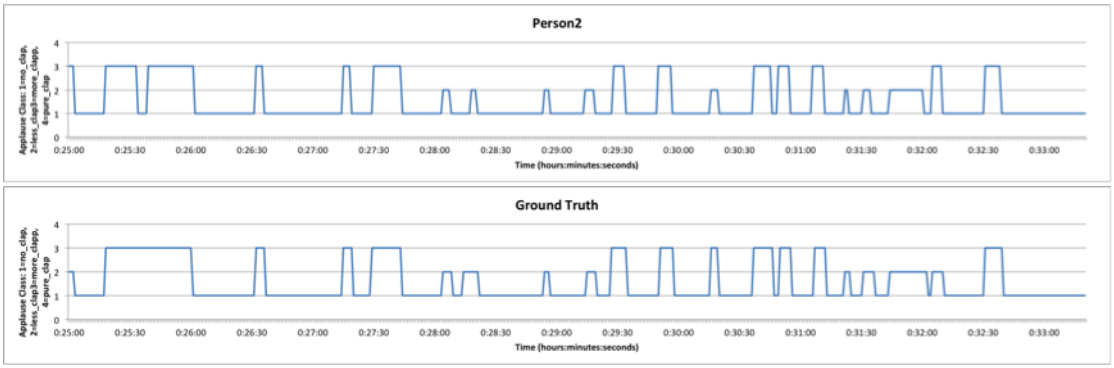


Figure 4.4: Comparison labelled applause data between ground truth and Person2

4.1.4 Evaluation

In order to evaluate the ground truth, we compare the ground truth with another person who manually labeled the applause sound on one video in the Circus Oz audio dataset. We do evaluation on one video because all videos on the dataset have the same type that is performance type video. This type of video has similar contents including: applause type, number of applause occurrences and duration. We called this person as Person2. Person2 is an IT graduate familiar with computer software but not involved in the project. This person independently did the labeling of the applause sound on the first of the videos labeled by Person1 that is audio_id: 01_devl, titled: Princes Park Melbourne. The process and the instructions for applause sound labeling are the same as for the ground truth. The only difference is that this person did not re-check the resultant applause clips.

Figure 4.4 illustrates the applause chart data between ground truth and Person2. The y-axis is the applause classes: 1=non clap, 2=less clap, 3=more clap, and 4=pure clap. It is compared with the manually labeled applause sound from 0.25:00 to 0.33.00 on audio id 01 devl. The Person2 applause data is almost the same as that of the ground truth. However, a couple of them are labeled as a different class. For example, at about 0.25.00, the ground truth is more clap although Person2 labeled it as less clap. It also noted that the timing code (start time and end time) is sometimes slightly different.

Kappa statistic is used to measure the agreement between ground truth and Person2. If they always agree, the Kappa value would be one. However, if they agree only at rate given

class	non_clap	less_clap	more_clap	pure_clap	total
no_clap	5,755	14	0	1	5,770
less_clap	0	211	25	2	238
more_clap	21	22	235	0	278
pure_clap	9	0	11	150	170
total	5,785	247	271	153	6,456

Table 4.5: Confusion matrix on duration (in seconds) of applause sound between Ground Truth and Person2.

by chance, the Kappa value would be zero. The Kappa value would be negative if they are worse than random.

Calculation of the Kappa value can be described as follows: First, the observed proportional agreement between two judgments X and Y is calculated:

$$p_o = \frac{1}{n} \sum_{i=1}^{g} f_{ii} \tag{4.1}$$

After that, the expected agreement by chance is calculated:

$$p_e = \frac{1}{n^2} \sum_{n=1}^{g} f_{i+} f_{+i} \tag{4.2}$$

where :

n = total number of clap

g = the number of class

f_{i+} = the total of ith row

f_{+i} = the total of ith column

Finally, the kappa statistic is:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{4.3}$$

audio_id	start_clap	end_clap	class
01_devl	224	226	2
01_devl	381	383	2
01_devl	435	443	2

Table 4.6: Sample of applause sound ground truth on CSV file format.

The Kappa statistic of agreement on the duration of applause between ground truth and Person2 can be calculated as follows:

$$p_o = \frac{1}{6,456} \times (5,755 + 211 + 235 + 150) = 0.983$$

$$p_e = \frac{1}{6,456^2} \times ((5,770 \times 5,785) + (238 \times 247) + (278 \times 271) + (170 \times 153)) = 0.804$$

$$\kappa = \frac{0.983 - 0.804}{1 - 0.804} = 0.916$$

According to the kappa value, we can conclude that there is good agreement between ground truth and Person2.

We made available the dataset for academic research purposes. The dataset and the applause sound ground truth can be downloaded[2]. The dataset contains the audio content only of the respective Circus Oz video. The ground truth data is stored in CSV file format containing the following fields: video id, clap start, clap end, class. The clap start and clap end value is the time (in seconds from start of video) the applause sound begins and the applause sound ends (Table 4.6).

4.2 Video segmentation dataset

This section describes the datasets associated with the technique of segmenting Circus Oz video into acts as explained in Chapter 6. There are four datasets that will be used to evaluate the proposed techniques for Circus Oz video segmentation. They are: single applause detection, black frames detection, images comparison and the end-of-detection datasets.

All datasets used in this section are taken from the collection of Circus Oz performance videos[3], as described in Section 4.1. Each dataset uses a different set of data appropriate for the goal of that proposed technique evaluation. For example, the dataset for the applause

[2]http://www.cs.rmit.edu.au/~jat/circusoz/
[3]http://archive.circusoz.com/

detection techniques is a set of applause sounds; for the black frames detection technique, it is a set of black frames. The details of each dataset are presented in the subsections below.

4.2.1 Single applause type dataset

The single applause dataset differs from the multiple applause type dataset described in Section 4.2. This single applause dataset consists of a certain type of applause data, while the multiple applause type dataset contains multiple applause types including less clap, more clap, and pure clap. Specifically, the audio class of this dataset is divided into two main classes: clap and non-clap. The clap class contains the sound of applause, while the non-clap class contains other sounds such as music, silence, voices and cheering.

Furthermore, the purposes of developing the two applause datasets are also different. On the one hand, the single-applause dataset is established to find the applause in a Circus Oz video. On the other hand, the multiple-applause dataset is developed in order to find the applause, and to differentiate the different types of applause. The single-applause dataset is used in a single-applause detection experiment in Section 6.2.2 that is part of the development of proposed end-of-act detection method.

In order to develop this dataset, we selected one performance circus video for each year from 1983 to 2012. This dataset contains 30 videos amounting to over 30 hours of viewing. The list of single-applause video dataset videos is shown in Table 4.7. Unfortunately, the majority of the videos are not available to public users. The reason is that most videos in the collection are not available to the public yet.

As this dataset will be used for finding the sound of applause on circus video experiment, we establish two main audio classes: clap and non-clap. The clap classes are those containing the applause sound, while the non-clap classes are those containing other sounds such as music, laughter, silence and speech. In the applause dataset development process, we searched short clips of each class and labeled these manually. For training and testing purposes, these short clips have a fixed 3-second duration as most applause is at least 3 seconds long. On each video, we took 30 sample clips divided into two classes. They are: 10 clap and 20 non-clap classes. There are 900 clips of 45 minutes duration in total.

No	Title	Video Access
1	1983 - Sydney, Australia, Sydney Entertainment Centre - 29 August	Non Public
2	1984 - Adelaide, Australia - Thebarton Theatre	Public
3	1985 - Albury, Australia, Big Top, Hovell Tree Reserve - 12 December	Public
4	1986 - Sydney, Australia, Big Top, Seymour Centre - 2 January	Non Public
5	1987 - Sydney, Australia, Big Top, Seymour Centre - 30 January	Non Public
6	1988 - Cairns, Australia, Cairns Civic Theatre - 6 July	Non Public
7	1989 - Melbourne, Australia, Melbourne Town Hall - 1 September	Public
8	1990 - Berkeley, USA, Zellerbach Hall - May	Non Public
9	1991 - Melbourne, Australia, Melbourne Town Hall - 14 September	Public
10	1992 - Chicago, USA, Blackstone Theatre - 14 June Matinee	Non Public
11	1993 - Gladstone, Australia, Gladston City Theatre - 22 October	Non Public
12	1994 - Copenhagen, Germany, Big Top, Tivoli Gardens - 21 July	Non Public
13	1995 - Bunbury, Australia, Bunbury Entertainment Centre - 10 March	Non Public
14	1996 - Hong Kong, China, Queen Elizabeth Stadium - 3 July	Non Public
15	1997 - Karratha, Australia, Outdoor Rig - 25 July	Public
16	1998 - Hobart, Australia, Theatre Royal - 12 March	Non Public
17	1999- Melbourne, Australia, Melbourne Town Hall - 26 June	Non Public
18	2000 - Melbourne, Australia, Melbourne Town Hall - 7 July	Non Public
19	2001 - Vienna, Austria, Ronacher Theatre - 4 November	Non Public
20	2002 - San Paulo, Brazil, Teatro Alfa - 1 June	Non Public
21	2003 - Ballarat, Australia, Big Top, Ballarat Botanical Gardens - 3 March	Non Public
22	2004 - Darwin, Australia, Darwin Entertainment Centre - 17 April	Non Public
23	2005 - Hamburg, Germany, Kampnagel - 8 December	Non Public
24	2006 - Wagga Wagga, Australia, Wagga Civic Centre - 6 May	Non Public
25	2007 - Pittsburgh, USA, Byham Theatre - 14 October	Non Public
26	2008 - Brighton, United Kingdom, Brighton Dome - 25 September	Non Public
27	2009 - Cardiff, Matinee, Sat,	Non Public
28	2010 Melbourne SoloBike	Non Public
29	2011 - Melbourne, Australia, Big Top, Birrarung Marr - 10 July	Non Public
30	2012 - Dubbo, NSW, Australia - 13 October	Non Public

Table 4.7: Video segmentation dataset

No	Title and URL	Duration	Video Access
1	1985 - Alice Springs, Australia, Araluen Arts Centre - 24 July, http://archive.circusoz.com/clips/view/36	01:45:12	Public
2	1990 - Richmond, Australia - Flying Fruit Fly Circus Big Top, Swan Street - 11 November, http://archive.circusoz.com/clips/view/48	01:41:06	Public
3	1993 - Alice Springs, Australia, Araluen Arts Centre - 18 September Matinee, http://archive.circusoz.com/clips/view/68	02:45:10	Non Public
4	1995 - Melbourne, Australia, Melbourne Town Hall - 4 October Matinee, http://archive.circusoz.com/clips/view/104	02:00:01	Non Public
5	1996 - New Delhi, India, Siri Fort, http://archive.circusoz.com/clips/view/195	01:29:50	Non Public
6	2000 - Sydney, Australia, Big Top, Moore Park - 1 January, http://archive.circusoz.com/clips/view/1	02:02:25	Non Public
7	2000 - Sydney, Australia, Big Top, Moore Park - 27 January, http://archive.circusoz.com/clips/view/3	01:43:30	Non Public
8	2000 - Melbourne, Australia, Melbourne Town Hall - 7 July, http://archive.circusoz.com/clips/view/5	01:31:19	Non Public
9	2003 - Sydney, Australia, Big Top, Moore Park, http://archive.circusoz.com/clips/view/321	01:15:04	Public
10	2005 - Brighton, UK, Brighton Dome - 17 November, http://archive.circusoz.com/clips/view/252	01:34:10	Non Public

Table 4.8: Image comparison dataset

4.2.2 Image comparison dataset

The image dataset is needed to evaluate the proposed image comparison technique. As proposed image comparison technique is used for evaluating applause sound, this dataset contains series of images taken from before and after applause occurred in Circus Oz videos.

The list of videos in the image comparison dataset is shown in Table 4.8. We selected 10 circus performance videos from the Circus Oz video collection. This dataset amounts to over 17 hours of viewing. Most videos are not available to public users.

In order to extract images from video, we established several parameters. Firstly, hue and saturation image features were used to generate an image histogram. Secondly, image resolution was set to the same resolution as a web video resolution, that is, 320 x 240

pixels. Finally, the image was taken 5 seconds before a clapping sound and 5 seconds after a clapping sound. This is because there are too many visual noises between 5 seconds before and 5 seconds after clapping sounds including camera operation (zooming and panning) and flashlights.

Two different types of datasets are used for the image comparison technique: single image and series of image comparison datasets.

Single image comparison dataset

The single image comparison dataset consists of two images on every applause sound that occurs in the video. That is an image taken at 5 seconds before the applause sound and another image taken at 5 seconds after the applause sound.

Series of image comparison dataset

Second, the series of image comparison datasets consist of 10 images for every applause sound occurring in the video. This series of images is divided into two: 5 images before the applause sound and 5 images after the applause sound. The first 5 images are extracted at every second from 5 to 10 seconds before the applause sound; the second 5 images are extracted at every second from 5 to 10 seconds after the applause sound.

4.2.3 Black frame dataset

The black frame dataset is used to test the black frame detection technique. The dataset contains the first 420 videos that have been uploaded on the Circus Oz video site. These videos are mostly performance videos taken from 1985 to 2004. The duration of the videos ranges from 40 seconds to three hours.

4.2.4 End-of-act dataset

The end-of-act dataset is used to evaluate the proposed end-of-act detection method. We selected ten Circus Oz videos from 1985 to 2005 for the end-of-act video dataset. The video dataset for the end-of-act detection method is listed in Table 4.8. This video dataset contains

ten Circus Oz videos totaling more than 17 hours of viewing time. As all videos in this dataset are performance type videos, the videos are quite lengthy. They are at least one hour long and a few of them are more than two hours long. Unfortunately, most of these videos in this dataset are not accessible to public users. Hence, we are unable to publish this dataset online as multiple applause type datasets in Section 4.2. Most of these videos were recorded in Australia and one was recorded in India and one in the UK.

We manually set the end-of-act time ground truth of every act on each video in the dataset. As we are aiming to evaluate the proposed end-of-act detection, the ground truth for this dataset is the end time of acts. In order to determine the end time of acts, we watched the Circus Oz video through a video player and wrote down the time that each act ended. The clues to end-of-act can be found by exploring its audio and visual contents. In visual content, the clue for end-of-act can be seen once the performers leave the stage or the stage lighting dims. In audio content, the clue for end-of-act can be when the audience applauds or the music stops. Once the end-of-act is found, we manually label the act title as Act + sequential act number, for example, Act 01.

Table 4.9 shows a sample of the list of acts and their end times for video dataset ID 05 (1996 - New Delhi, India, Siri Fort). We search the time when act ends and manually label the name of the act. There are 11 acts in this video. The first act, Act 01, ends at 0:02:42 and the second act, Act 02, ends at 0:04:45. According to the end-of-act time data, the duration of Act 01 is 2 minutes and 42 seconds while the duration of Act 02 is 2 minutes and 3 seconds.

We searched and labeled each act and its end-of-act time for all 10 videos in the dataset. The number of acts for each video is shown in Table 4.10. There is a total of 195 acts for all 10 videos in the dataset, and an average of 19 acts per video.

As the proposed end-of-act detection method involves three other techniques, applause detection, black frames detection, and image comparison techniques, the dataset is also divided into three parts: applause, black frame, and image dataset. The development of these datasets has been described in Section 4.3.1, 4.3.2, and Section 4.3.3.

Act No	End-of-act time
Act 01	0:02:42
Act 02	0:04:45
Act 03	0:16:51
Act 04	0:22:48
Act 05	0:29:06
Act 06	0:36:12
Act 07	0:54:45
Act 08	1:00:03
Act 09	1:14:27
Act 10	1:20:48

Table 4.9: List of acts and its end_time for video dataset ID 05 (1996 - New Delhi, India, Siri Fort)

dataset ID	Number of acts
01	22
02	20
03	30
04	18
05	11
06	24
07	20
08	13
09	16
10	21

Table 4.10: The number of acts on end-of-act detection dataset

Applause component of end-of-act dataset

The applause component for end-of-act detection is generated using the proposed single-applause detection technique described in Chapter 6. The applause dataset for the end-of-act detection experiment is the two classes dataset: clap and non-clap. We applied our applause detection technique in order to obtain the start time and end time of the applause in each video.

Table 4.11 shows a sample of the applause sound detected on dataset ID:01, 1985 - Alice Springs, Australia, Araluen Arts Centre - 24 July. In this video, 72 applause sounds are detected. Forty of these are listed in this table. For example, Clap 01 occurred from 0:10:13 to 0:10:15 with 2 second duration. The duration of the applause varies from at least 2 seconds to more than 14 seconds.

We applied the proposed applause detection technique to ten end-of-act videos dataset. The detected applause is then used to evaluate the end-of-act detection method. Table 4.12 shows the statistics of manual labeling applause and its duration. As we can see from the table, the total applause for all ten video datasets is 964 applause sounds. The average number of applauses per video is 96.

Table 4.12 also shows the duration of the applause sounds detected on ten end-of-act videos dataset. As we can see from the graph, the total duration of applause sounds detected in all ten video datasets is an hour and 20 minutes. The average applause duration per video is more than eight minutes. The longest applause duration is 11 minutes and 17 seconds on dataset ID 04 and the shortest applause duration is three minutes 11 seconds on dataset ID 05.

Image component of end-of-act dataset

An image component of the dataset is also needed to evaluate detected applause and whether or not the applause occurs at the end-of-act. In a circus performance, the audience often applauds both in the middle and at the end of an act. In order to distinguish applause in the middle of act from applause at the end of the act, the image taken from before the applause and the image taken after the applause are compared.

Clap No	Start Time	End Time	Duration
Clap_01	0:10:13	0:10:15	02
Clap_02	0:12:31	0:12:33	02
Clap_03	0:15:13	0:15:24	11
Clap_04	0:16:01	0:16:03	02
Clap_05	0:16:49	0:16:51	02
Clap_06	0:21:10	0:21:15	05
Clap_07	0:22:22	0:22:24	02
Clap_08	0:22:31	0:22:33	02
Clap_09	0:23:43	0:23:51	08
Clap_10	0:26:55	0:27:09	14
Clap_11	0:28:46	0:28:48	02
Clap_12	0:28:52	0:28:54	02
Clap_13	0:29:19	0:29:24	05
Clap_14	0:31:13	0:31:18	05
Clap_15	0:31:37	0:31:42	05
Clap_16	0:32:49	0:32:51	02
Clap_17	0:33:31	0:33:42	11
Clap_18	0:34:22	0:34:27	05
Clap_19	0:35:01	0:35:03	02
Clap_20	0:37:46	0:37:48	02
Clap_21	0:38:52	0:38:57	05
Clap_22	0:39:40	0:39:45	05
Clap_23	0:41:19	0:41:27	08
Clap_24	0:42:04	0:42:06	02
Clap_25	0:43:07	0:43:12	05
Clap_26	0:46:52	0:46:57	05
Clap_27	0:50:31	0:50:33	02
Clap_28	0:50:37	0:50:42	05
Clap_29	0:51:19	0:51:24	05
Clap_30	0:52:13	0:52:21	08
Clap_31	0:54:34	0:54:36	02
Clap_32	0:54:55	0:54:57	02
Clap_33	0:55:46	0:55:51	05
Clap_34	0:57:34	0:57:36	02
Clap_35	0:58:01	0:58:06	05
Clap_36	0:59:31	0:59:36	05
Clap_37	1:00:40	1:00:45	05
Clap_38	1:02:19	1:02:21	02
Clap_39	1:03:04	1:03:09	05
Clap_40	1:04:34	1:04:36	02
...	...	...	...

Table 4.11: Detected applause sound on dataset ID 01

dataset ID	Number of applause	Duration of applause
01	72	0:06:06
02	111	0:09:38
03	156	0:11:06
04	130	0:11:17
05	46	0:03:11
06	105	0:08:39
07	87	0:06:42
08	79	0:07:17
09	69	0:05:48
10	109	0:10:36

Table 4.12: The manually detected number of applause and their duration on end-of-act video dataset

dataset ID	Number of single image	Number of series of images
01	114	720
02	222	1110
03	312	1560
04	260	1300
05	92	1300
06	210	1050
07	174	870
08	158	790
09	138	690
10	218	1090

Table 4.13: Single image and series of images dataset

There are two image datasets: single and series of image dataset. Both datasets have been generated as described in Section 4.3.2. The single image dataset contains two images for each applause, whereas the series of image dataset contains ten images for each applause that are five images taken before the applause and other five images taken after the applause. Based on the number of applauses per video in (Table 4.12),the statistics of single image and series of image dataset are displayed in Table 4.13.

Black Frame ID	Start Time	End Time	Duration
Black_01	0:19:05	0:19:08	03
Black_02	0:23:33	0:23:35	02
Black_03	0:30:18	0:30:30	12
Black_04	0:31:47	0:31:50	03
Black_05	0:33:13	0:33:15	02
Black_06	0:41:35	0:41:40	05
Black_07	0:46:51	0:46:54	03
Black_08	0:55:35	0:55:37	02
Black_09	0:56:45	0:56:52	07
Black_10	1:00:14	1:00:25	11
Black_11	1:14:22	1:14:25	03
Black_12	1:18:56	1:18:59	03
Black_13	1:27:22	1:27:26	04
Black_14	1:31:25	1:31:29	04
Black_15	1:34:20	1:34:27	07

Table 4.14: Detected black frame on dataset ID 02

Black frame component of end-of-act dataset

The black frame component of the dataset is needed to investigate whether or not black frames occur near the applause sound. The black frame dataset is extracted from the end-of-act dataset using ffmpeg multimedia tool. We set the value for minimum black frame duration to two seconds, as most black frames duration on act transition in Circus Oz video is at least two seconds. As the circus performers present their shows mostly in dim light, the black ratio value was set to 98%.

An example of detected black frames for dataset id 02 (1990 - Richmond, Australia - Flying Fruit Fly Circus Big Top, Swan Street - 11 November, http://archive.circusoz.com/clips/view/48), is listed on Table 4.14. Once the black frames are detected, we store the start time and time of the black frame in a database titled Black + sequential number. For example, the first black frame, Black 01, occurred from 0:19:05 to 0:19:08 with a three-second duration.

The black frames detection technique is employed to obtain the start time and end time of black frames on the end-of-act video dataset. The number of black frame segments and their duration is shown in Table 4.15. The total number of black frames on ten video datasets is

dataset ID	Number of blackframe segments	Duration of blackframe
01	14	0:04:23
02	15	0:01:11
03	18	0:02:58
04	8	0:01:06
05	6	0:01:53
06	36	0:06:24
07	40	0:11:25
08	15	0:01:44
09	18	0:01:58
10	27	0:05:25

Table 4.15: The detected blackframes and their duration on end-of-act video dataset

197, and the total duration is 38 minutes and 27 seconds. The average number of black frame segments per video is 19 black frames. The highest black frames number is 40 black frames on video dataset ID 07, whereas the lowest black frames number is six on video dataset ID 05. Furthermore, the average black frame duration is three minutes and 51 seconds. The longest black frames is 11 minutes and 25 seconds on video dataset ID 07 while the shortest black frames duration is a minute and 6 seconds on video dataset ID 04.

4.3 Summary

The multiple applause type dataset and video segmentation dataset have been developed. These datasets are used to test the performance of the proposed applause sound detection and video segmentation technique. Below, we highlight the following main points associated with the development of the dataset.

Multiple applause type dataset. The multiple applause type dataset comprises 12 video Circus Oz performance videos totaling 21 hours, 19 minutes and 53 seconds of viewing. This dataset is used to evaluate proposed multiple applause type detection technique in Chapter 5.

Video segmentation dataset. The video segmentation dataset has been set up. It comprises ten video Circus Oz performance videos totaling more than 17 hours of viewing. In

Chapter 6, this dataset is used to evaluate the proposed technique for the segmentation of a video into acts.

Chapter 5

Applause Detection Technique

Two approaches can be used as applause detection techniques: characteristic-based and classification-based. Several audio features are explored including: energy, spectral, MFCC and PLP. Furthermore, the Circus Oz applause data set explained in Chapter 4 is used to evaluate the performance of both characteristic-based and classification-based applause detection techniques.

5.1 Characteristic-based approach

5.1.1 Method

One way to detect the sound of applause is by using a characteristic-based approach that finds the unique audio feature characteristics of the applause sound that distinguish applause from other sounds. Several researchers proposed the applause detection technique using the characteristic-based approach. For example, Li et al. [2009] used duration, pitch, spectrogram, and occurrence location to detect the applause sound on an audio recording of a meeting. Similarly, Manoj et al. [2011] proposed characteristic-based applause detection on an audio recording of a meeting speech. They used the following audio features: decay factor, lag of first minima of ACF, index of FFT bins for BER and band energy ratio. Another characteristics-based approach is proposed by Sarala et al. [2012]. They used spectral audio features to detect the applause sound in a Carnatic music concert.

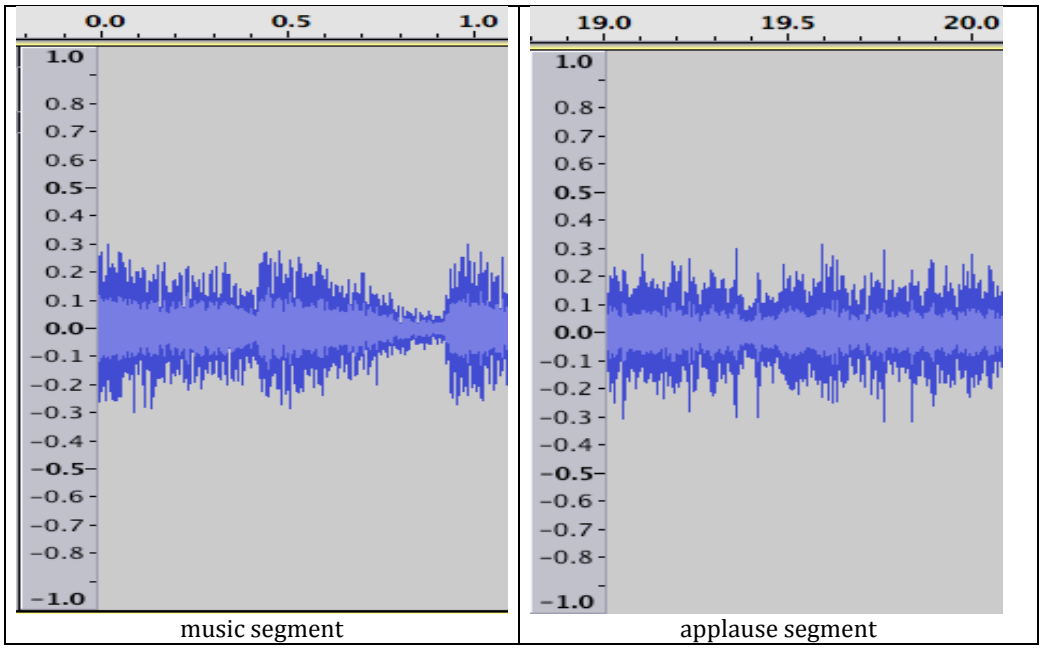


Figure 5.1: Time domain wav figure on music segment on the left and applause segment on the right side

For applause dataset implementation, Li et al. [2009] and Manoj et al. [2011] used the audio recording of a meeting speech, while Sarala et al. [2012], CUSUM technique, used a Carnatic music concert. Detecting applause sound in a meeting speech is basically distinguishing between applause sound and speech. Detecting the sound of applause in a music concert requires distinguishing audio characteristic music from the applause sound.

As the audio content of a Circus Oz data set is mostly music, we used Sarala et al. [2012]'s approach for detecting applause in a Circus Oz performance video archive. Sarala et al. [2012] analyzed the audio signal changes between music and applause. The difference between music and applause segments in the time domain and the spectral domain can be seen in Figure 5.1, Figure 5.2 and Figure 5.3. As can be seen in those figures, a music segment spectrum fluctuates where an applause segment spectrum is flat.

The steps for detecting applause sound as described in Sarala et al. [2012] are as follows: First, the spectral audio features are extracted every 23ms. The extracted spectral values are then smoothed and normalized. Then, the cumulative sum (CUSUM) formula is applied

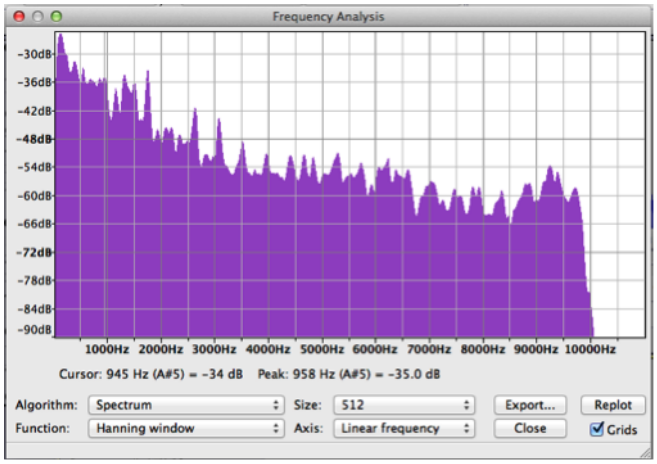


Figure 5.2: Spectral entropy on music segment

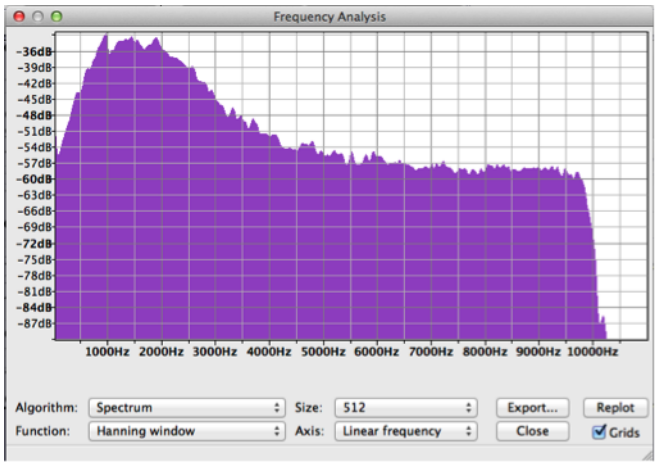


Figure 5.3: Spectral entropy on applause segment

to detect the level and duration of the sound of applause. The applause is detected if the CUSUM value is greater than 0; otherwise the sound is not the sound of applause.

The spectral value is smoothed using moving average filter formula [Smith, 1997] as follows:

$$Y[i] = \frac{1}{M} \sum_{j=0}^{M-1} X[i+j] \tag{5.1}$$

where $Y[i]$ is the smoothed spectral value of the ith frame of an audio signal, M is the filter size, and X[i] is the spectral value of the ith frame of an audio signal.

The spectral value is also normalized using peak normalization formula as it perfoms best compared to no normanlization and power spectal density normalization [Sarala et al., 2012]:

$$XNorm_n(\omega) = \frac{X_i(\omega)}{max_\omega X_n(\omega)} \tag{5.2}$$

where $XNorm_n(\omega)$ is the normalized power spectrum of the nth frame and X_n is the power spectral value of the nth frame of an audio signal.

Figure 5.4 shows the change in the time domain from the music to the applause segment on Circus Oz video. The audio signal on the left of the vertical line is music, while the audio signal on the right of the vertical line is the sound of applause. The spectral values are extracted from both audio signals as shown in Figure 5.5. After that, the spectral flux values are normalized and smoothed using the formula above. The result of the normalized and smoothed spectral flux value can be seen in Figure 5.6. Finally, the CUSUM formula is calculated. As shown in Figure 5.7, the applause sound is detected if the CUSUM value is greater than 0.

We conduct several experiments where we apply the CUSUM technique to the Circus Oz video archive. These are described as follows.

First, having defined the three applause classes in our applause data set, we want to conduct several experiments to measure the CUSUM value changes in terms of those applause

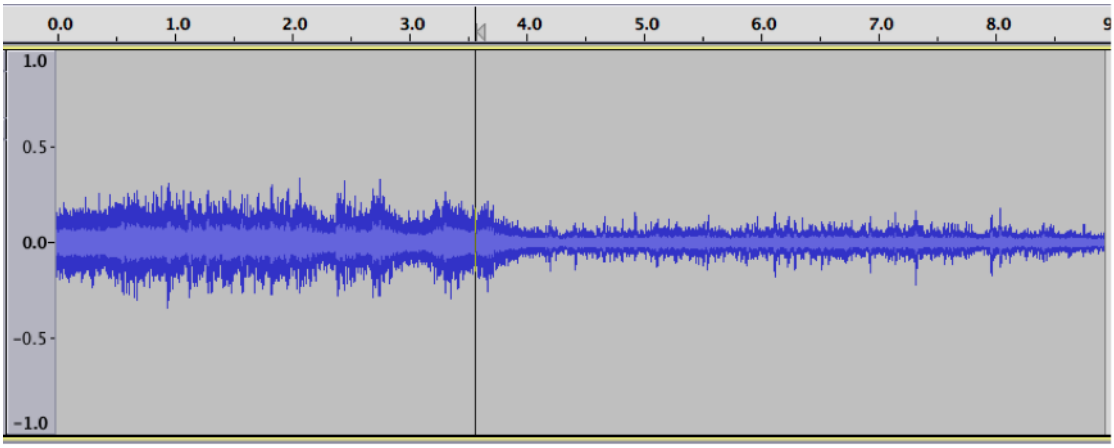


Figure 5.4: Changing between music and applause segments

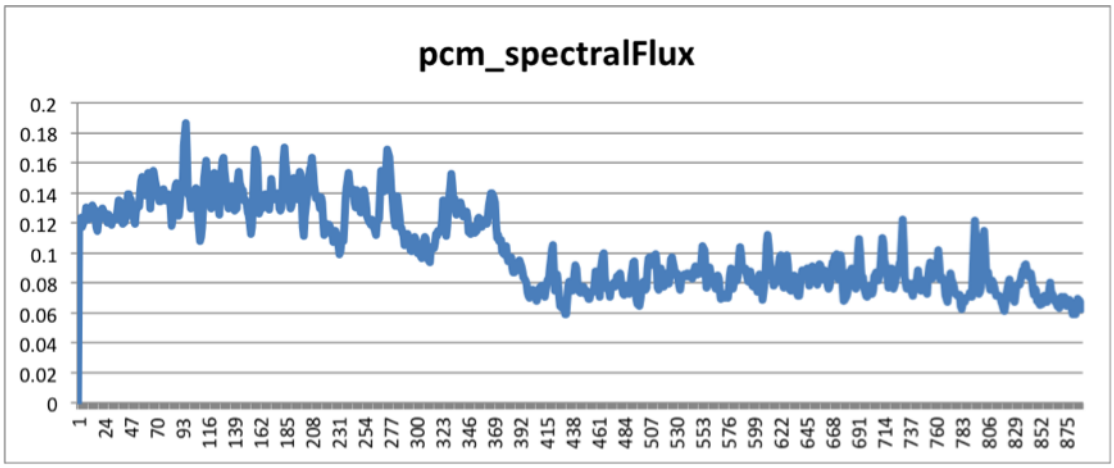


Figure 5.5: Spectral flux changing between music and applause segments

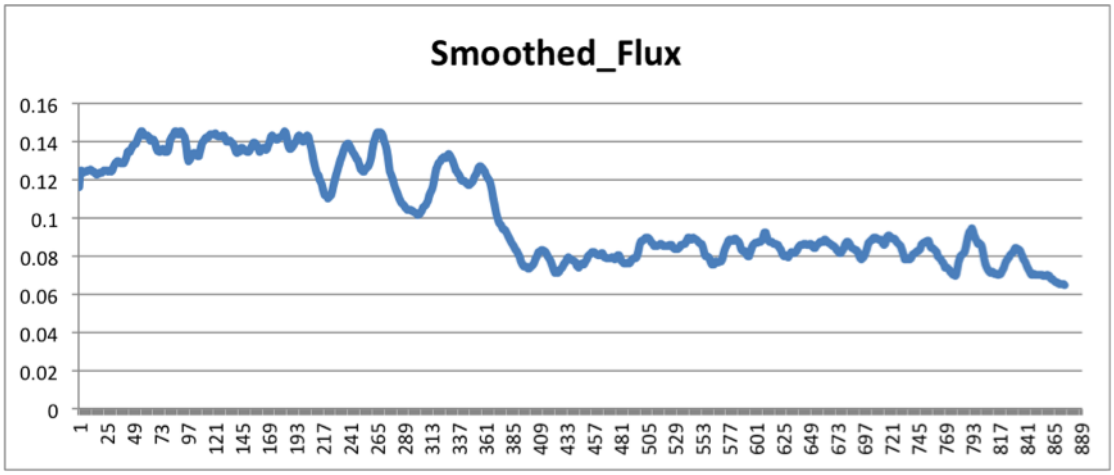


Figure 5.6: Normalized and smoothed spectral value

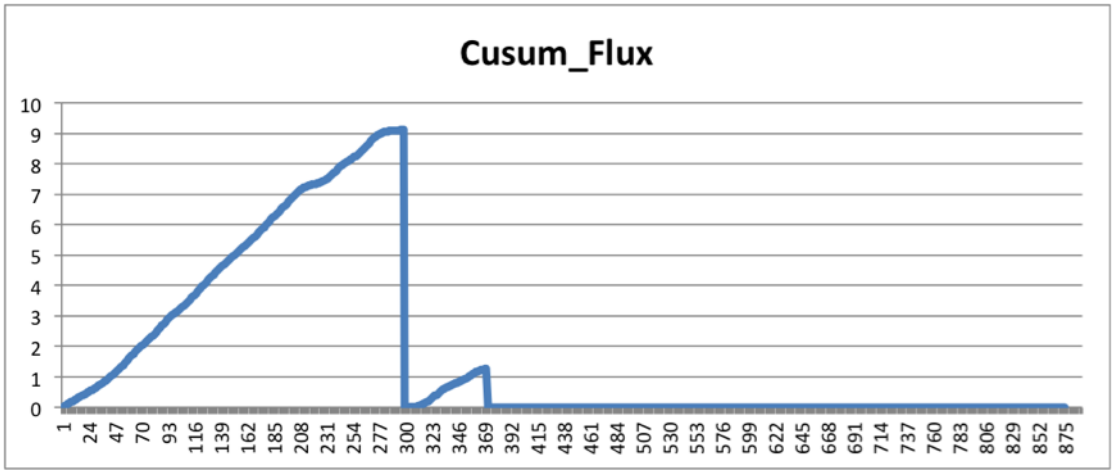


Figure 5.7: CUSUM value from spectral flux changes between music and applause

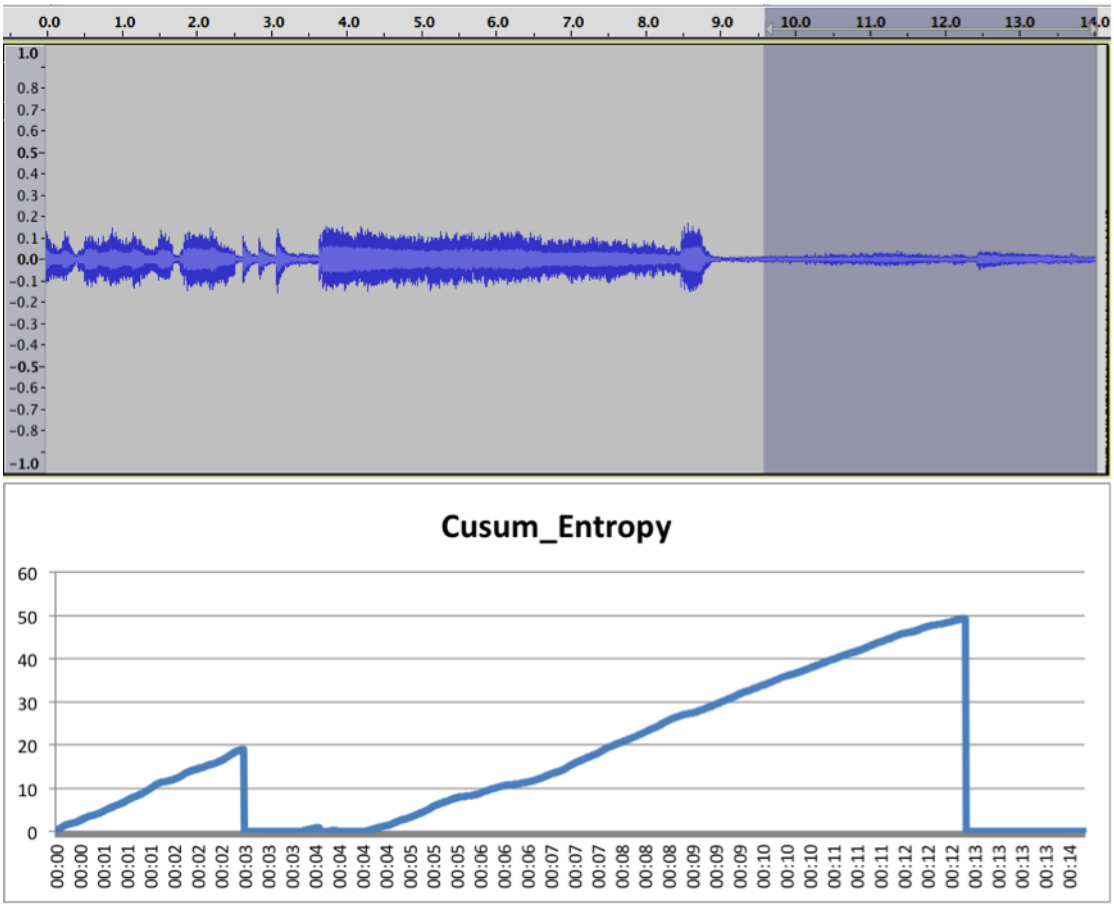


Figure 5.8: CUSUM value changing from music to less_clap

classes. The CUSUM value changes from music to low clap, more clap, and pure clap applause classes can be seen in Figure 5.8, Figure 5.9, and Figure 5.10. As we can see from those figures, the CUSUM algorithm works well in distinguishing music and pure clap. However, it is much more difficult to detect less clap and more clap.

Next, in the Circus Oz video, the applause sound not only occurs after music as in a Carnatic music concert; it can also appear after other sounds such as cheers, speech, silent, and laughing. The following figures illustrate the CUSUM value changes from different sounds to applause sounds. Figure 5.11 shows CUSUM value changes from speech to the applause sound. Figure 5.12 shows CUSUM value changes from speech and then laughter and applause sound. Figure 5.13 shows CUSUM value changes from music – applause –

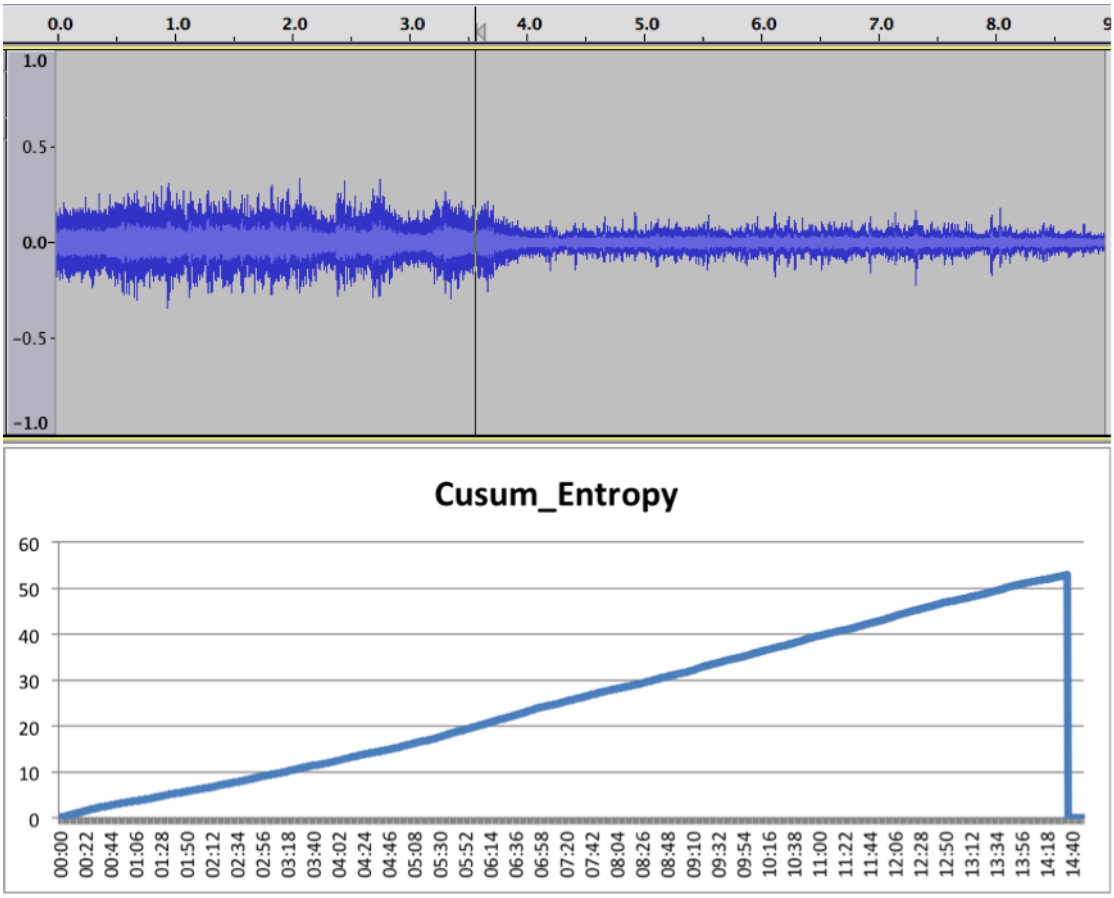


Figure 5.9: CUSUM value changing from music to more_clap

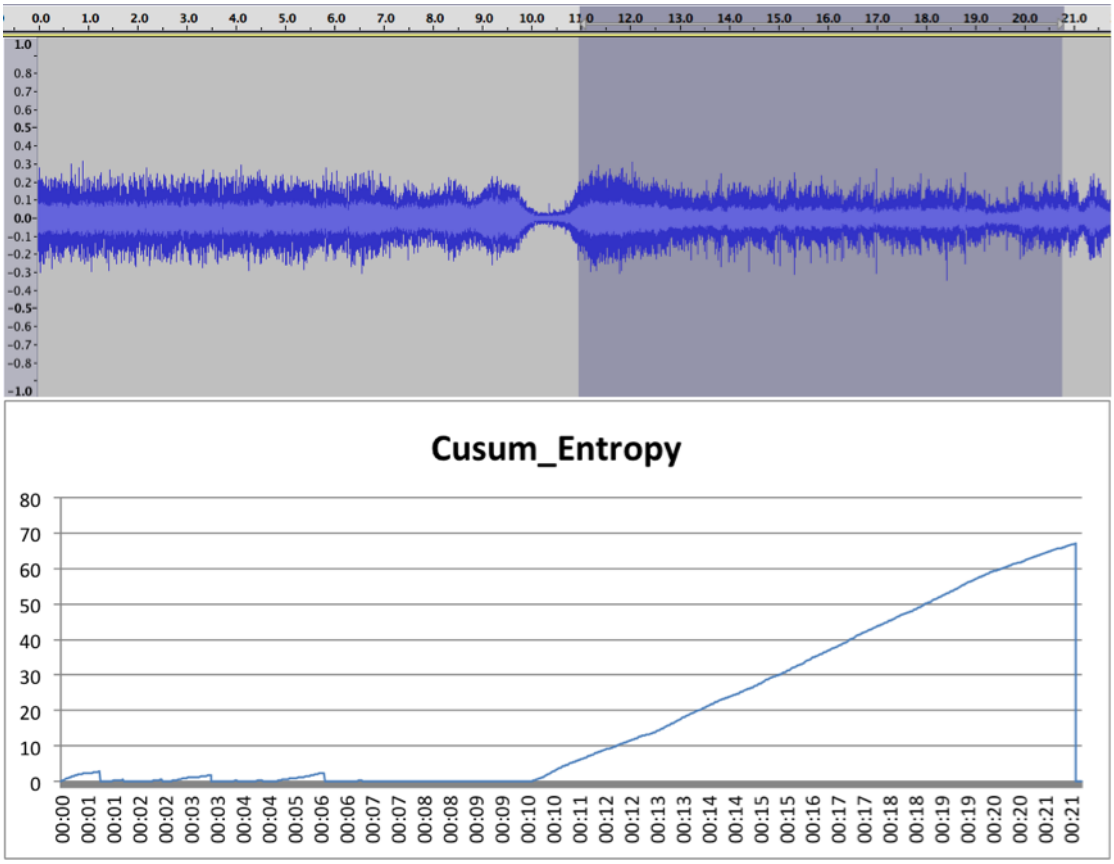


Figure 5.10: CUSUM value changing from music to pure_clap

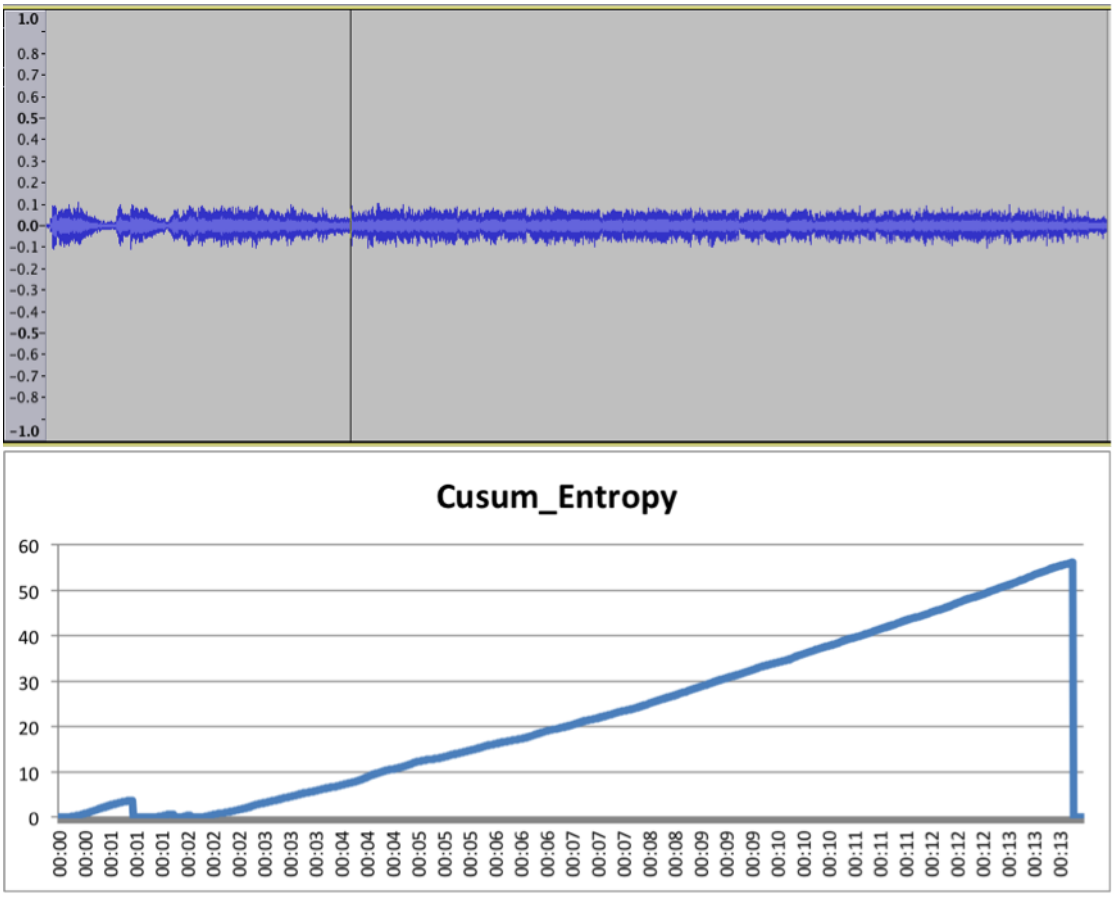


Figure 5.11: CUSUM value changing from speech to applause

speech sounds. Although the applause segment is not quite clearly detected, we can still see where the applause sound occurs.

Next, in order to extract the spectral entropy for each frame, we use two calculation-based approaches: power spectrum and magnitude spectrum. Figure 5.14 shows the CUSUM value changing from music to pure clap based on the power and magnitude of the spectrum calculation. Figure 5.15 shows the CUSUM value changing from music to pure clap to more clap based on the power and magnitude of the spectrum calculation. Both figures show that the magnitude spectrum-based calculation performs better.

Finally, audio signal smoothing is another important aspect of applause sound detection. The audio signal is smoothed using an average filter function. Two things must be considered when calculating the average filter function. They are: filter size and symmetrical or

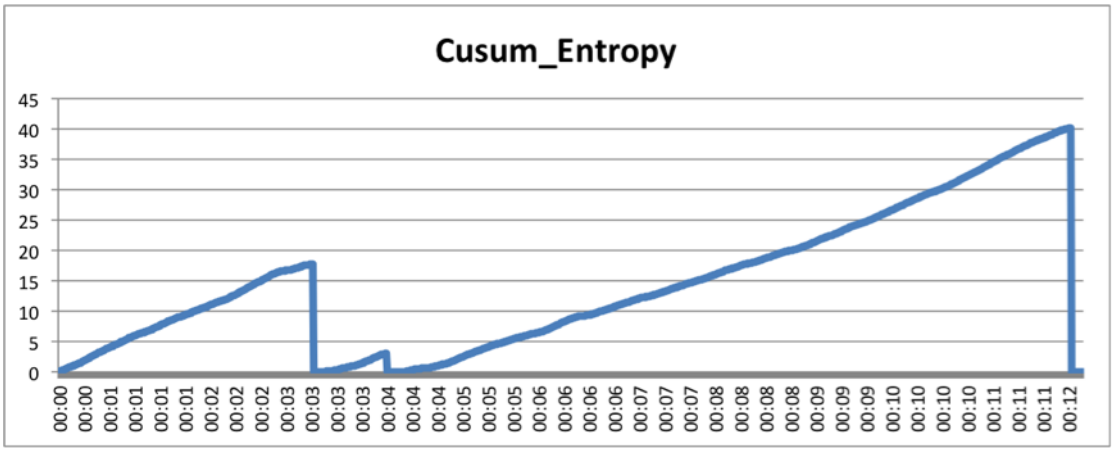


Figure 5.12: CUSUM value changing from speech - laugh - applause

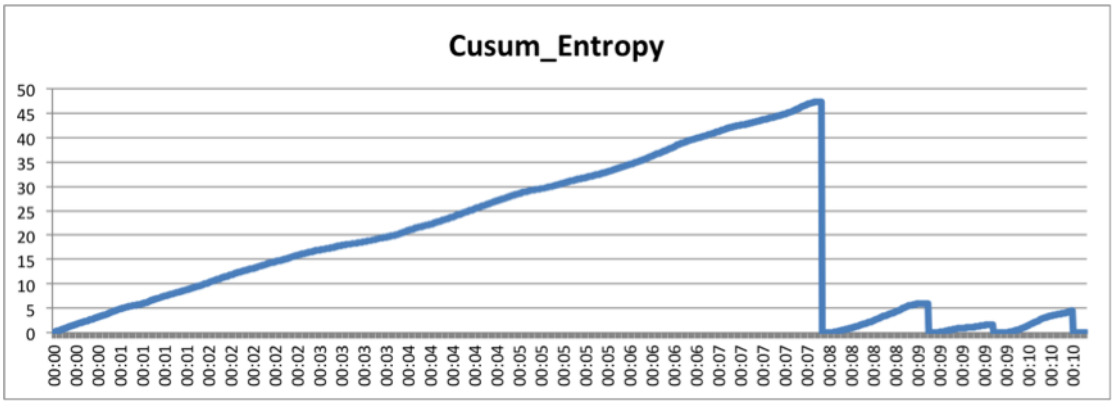


Figure 5.13: CUSUM value changing from music - clap - speech

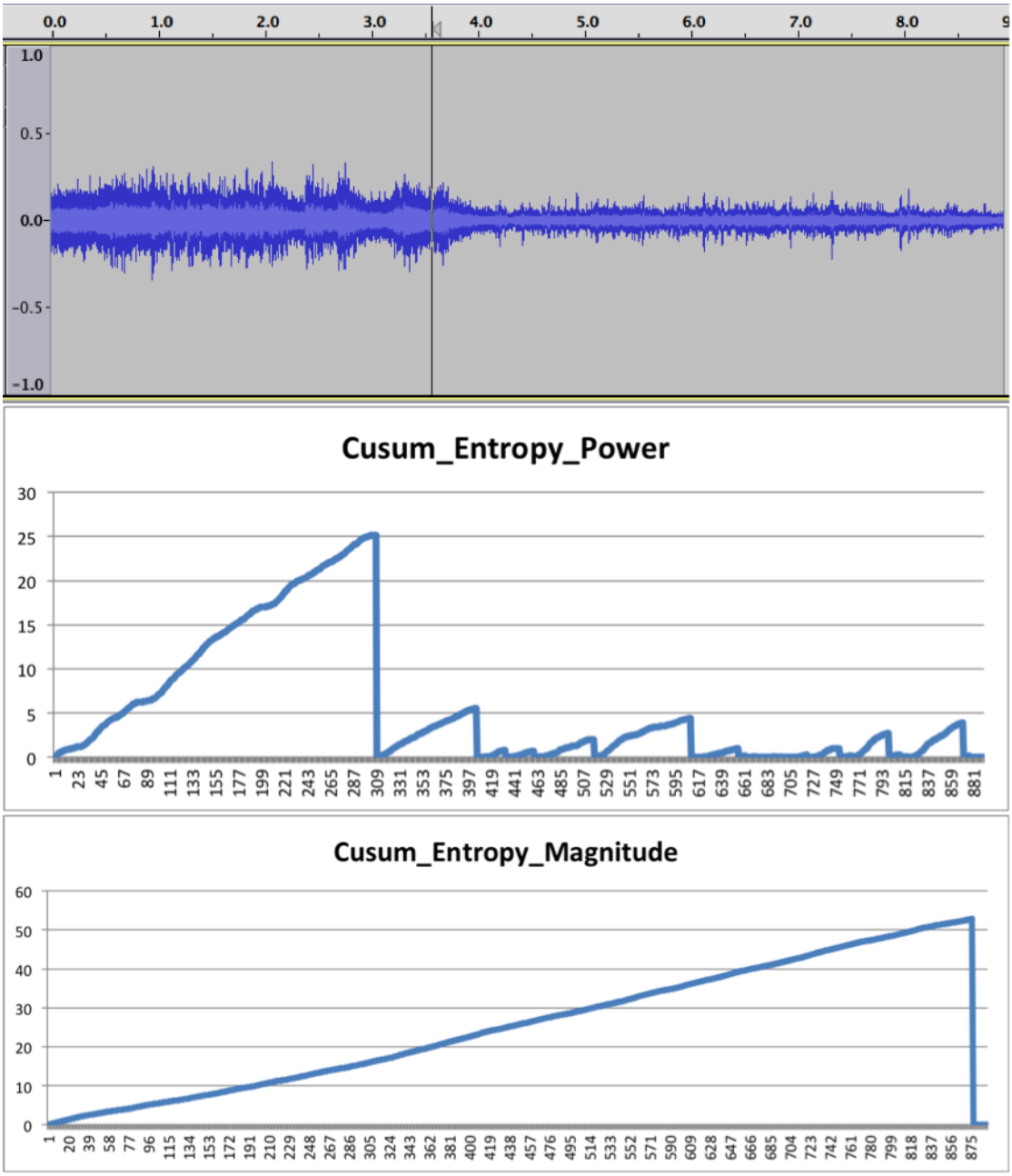


Figure 5.14: CUSUM value changing from music to pure_clap based on power and magnitude spectrum calculation

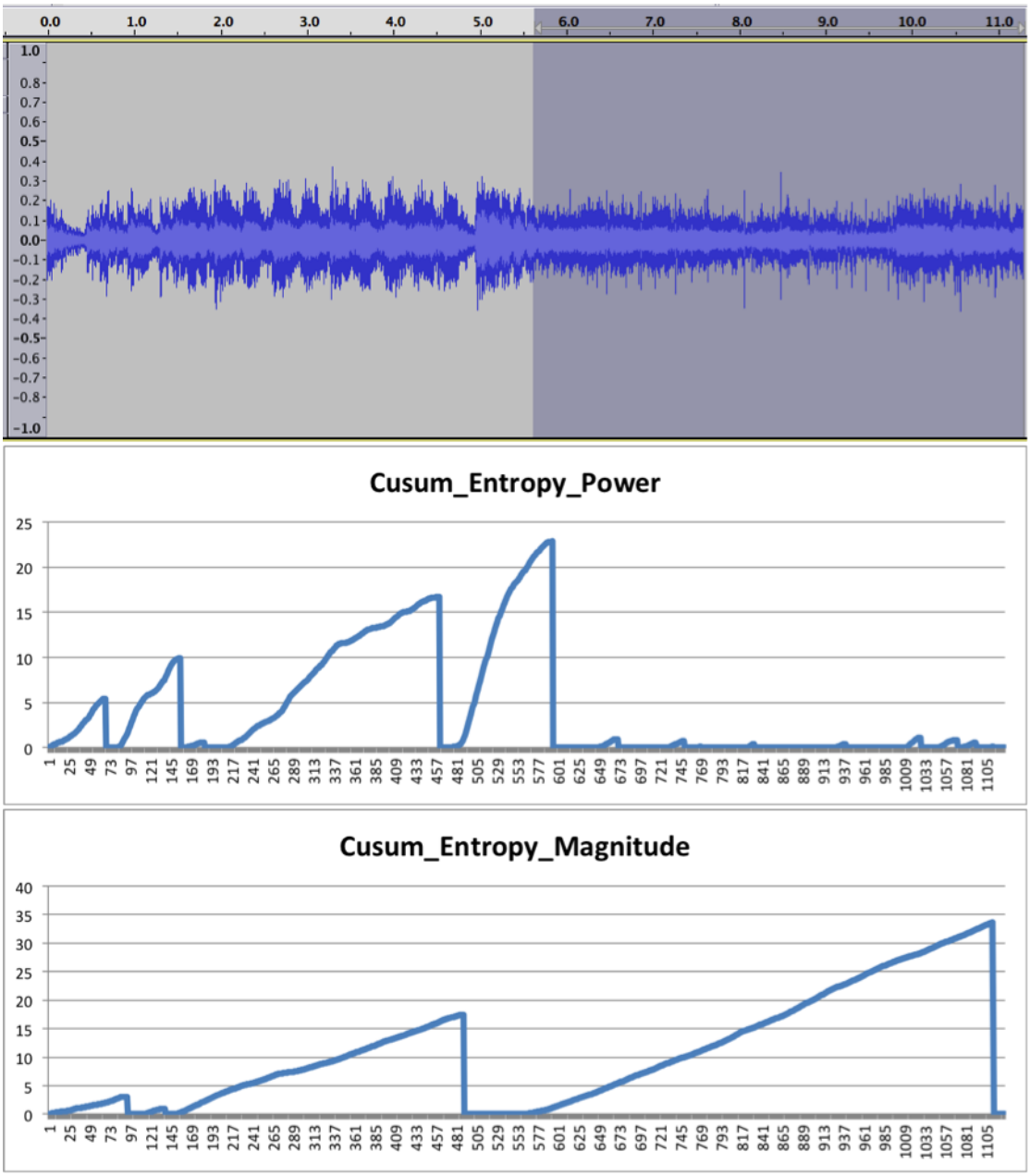


Figure 5.15: CUSUM value changing from music to pure_clap based to more_clap on power and magnitude spectrum calculation

asymmetrical calculation-based. The result of the experiment on the filter sizes is shown in Figure 5.16. The filter size is set to different sizes: 5, 10, 15, and 20 point moving average filter. Filter size 10 seems to be the best result. Figure 5.17 shows two average filter calculation approaches: symmetrical and asymmetrical. The asymmetrical calculation produces a better result.

According to the experiments above, we can conclude that the following parameters are suitable when implementing the CUSUM technique on the performance videos archive of Circus Oz:

- Frame size: 0.01 without overlapping
- Smoothing average filter size : 10 with no symmetric calculation
- Audio feature: spectral entropy
- Audio feature calculation: magnitude spectrum

These experiment parameters will be used to measure the success of applause detection using the CUSUM algorithm in sub-section 5.1.2 below.

5.1.2 Performance and evaluation

The applause sounds of ground truth data are classified into three classes: less clap, more clap, and pure clap. However, based on our experiment, the CUSUM technique can only detect applause or non applause sound. Therefore, in this experiment, we evaluate only the appearance of the sound of applause regardless of its class as defined in the ground truth data.

Evaluation approach

We use timing evaluation rather than clips evaluation. This is because in clips evaluation the start time and the end time of detected applause might be not exactly the same as the ground truth. Figure 5.18 illustrates five possibilities matching start time and end time between ground truth and CUSUM-detected applause sound. The CUSUM b and c definitely

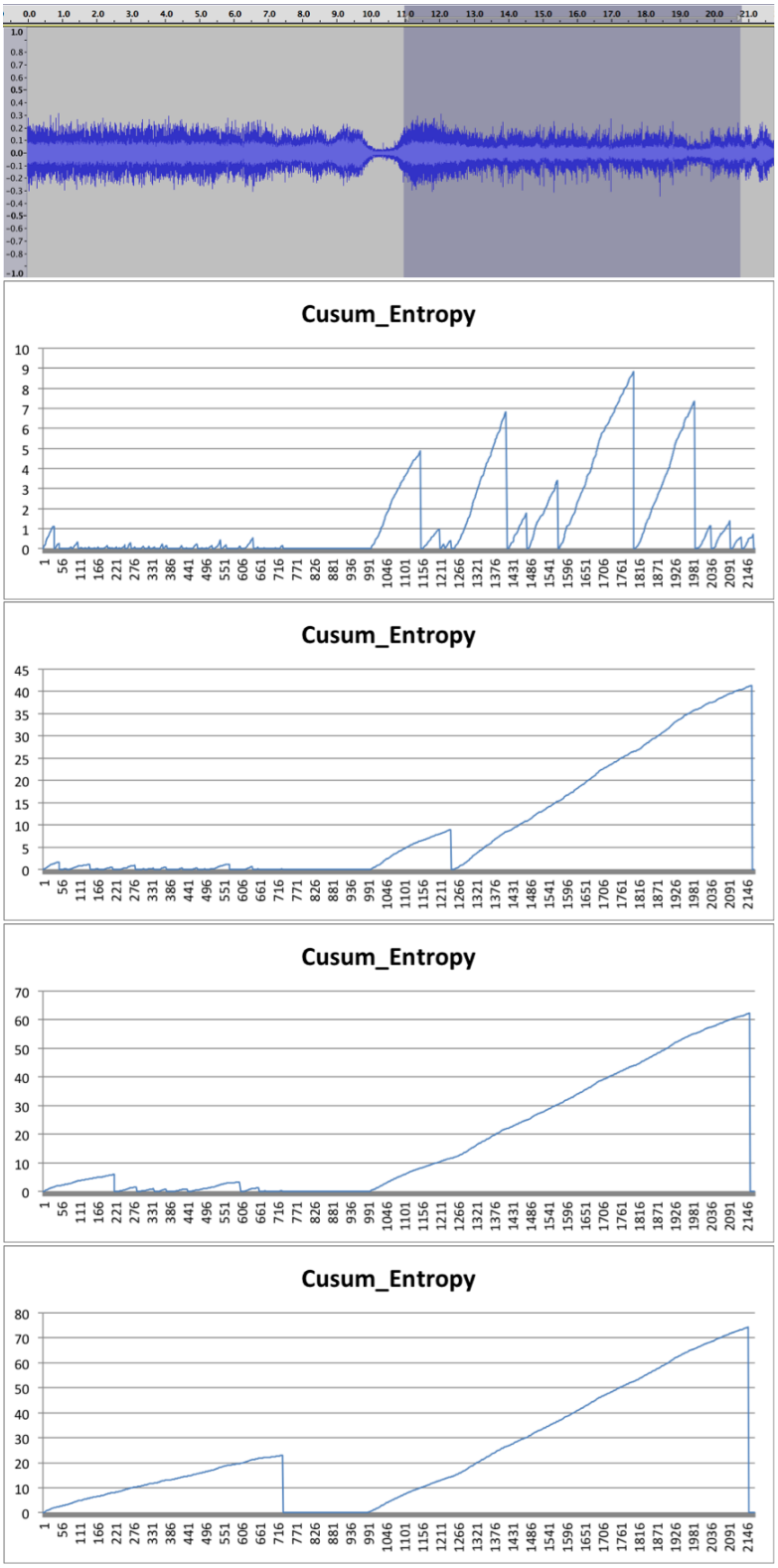


Figure 5.16: CUSUM value changing on different average filter size: 5, 10, 15, and 20

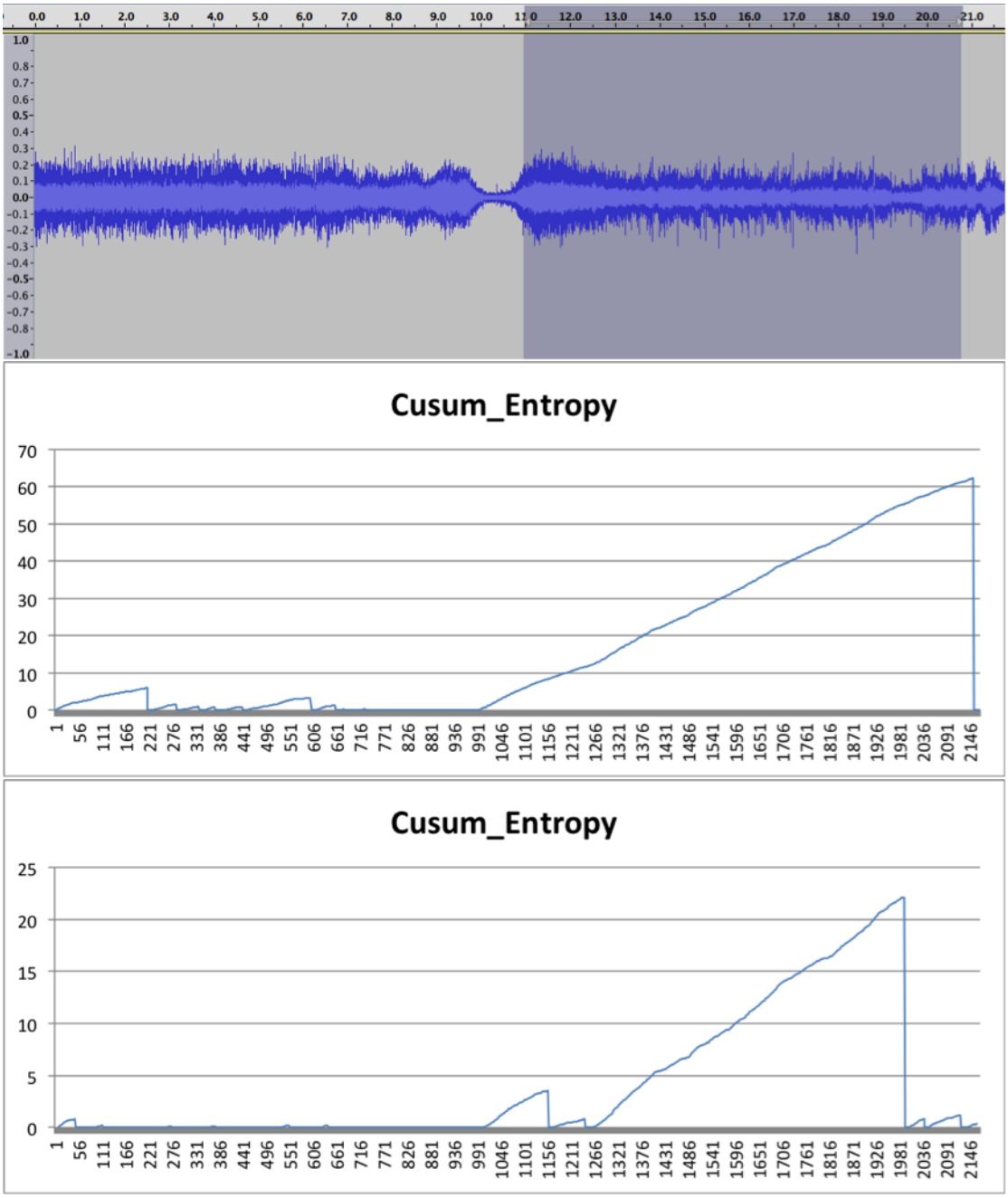


Figure 5.17: CUSUM value changing on different average filter calculation: asymetrical and symetrical

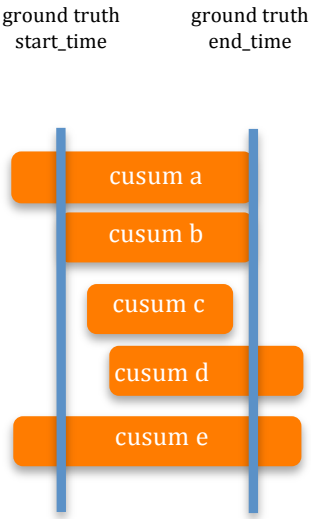


Figure 5.18: Matching posibilities between detected applause and ground truth

match with ground truth. However, other CUSUMs are not an exact match with the ground truth. For example, the end time of CUSUM matches the end time of ground truth, but the start time is earlier. Similarly, the start time of CUSUM d is within the start time of ground truth but the end time is greater than the ground truth. In addition, the start time and the end time of CUSUM e are less and greater than the ground truth respectively.

On timing evaluation, every second of detected applause sound will be compared with ground truth data. First, all applause clips from ground truth are framed into a second of clip. After that, each second of clip is labeled according to the class of ground truth. Given that data, we can then generate every second of non-clap data. This process can be illustrated in Figure 5.19.

Second, similar to the ground truth process, all detected applause clips are framed into a second of clip. As the frame size of the audio feature extracted with the CUSUM approach is 0.01ms, the durations of framed clips are sometimes not exactly a second. Therefore, we use the round() function to make the frame become one second. Furthermore, this clip is only labeled as applause clip. Given that data, we can also generate the non-applause data. This process is illustrated in Figure 5.20.

Finally, we can compare each second of frame between applause data on ground truth

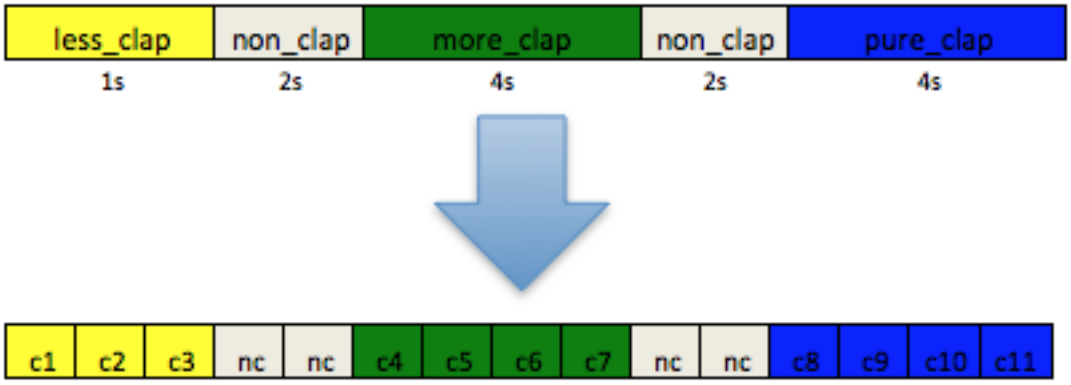


Figure 5.19: Framed ground truth applause sound data

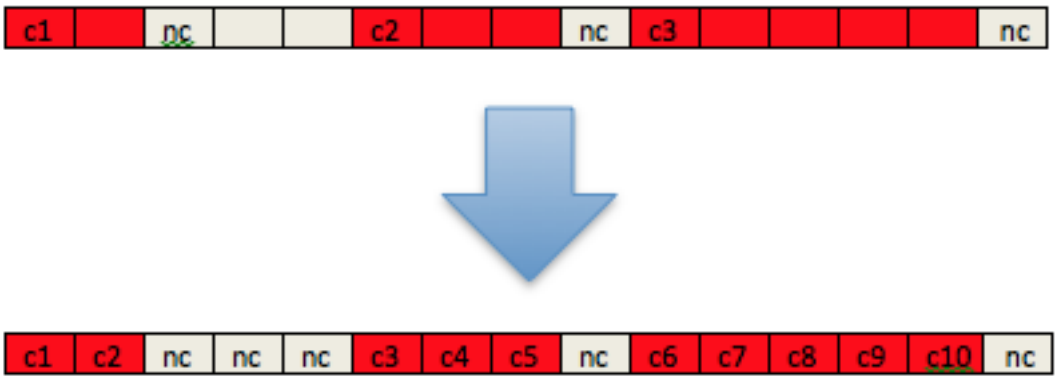


Figure 5.20: Framed ground truth applause sound data

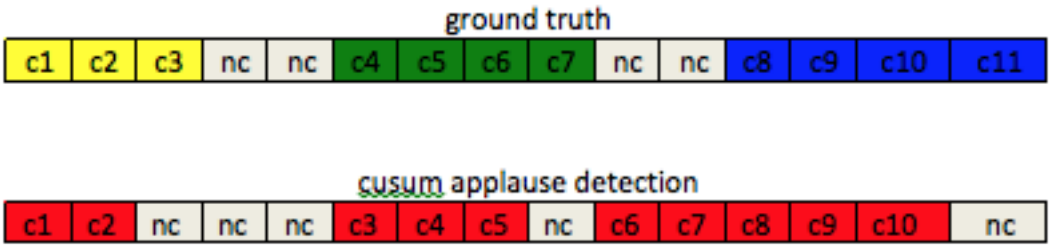


Figure 5.21: Compare a second applause ground truth and CUSUM data

and CUSUM technique as shown in Figure 5.21. The following detected CUSUM applauses are matched with ground truth: c1, c2, c3, c4, c5, c8, c9, and c10. Other detected CUSUM applauses are not matched. They are: c6 and c7.

Precision and recall calculation

After all the clips being labelled, we use the following formula to calculate precision and recall:

$$precision = \frac{tp}{tp + fp} \tag{5.3}$$

$$recall = \frac{tp}{tp + fn} \tag{5.4}$$

$$F = 2 \times \frac{precision \times recall}{precision + recall} \tag{5.5}$$

where:

tp=true positive

fp=false positive

fn=false negative

		ground truth	
		Condition positive	Condition negative
cusum	Test outcome positive	*tp*=8	*fp*=2
	Test outcome negative	*fn*=3	*tn*=2

Figure 5.22: Example of confusion matrix

For example: the confusion matrix from Figure 5.21 can be calculated as shown in Figure 5.22.

The precision, recall and F value can be calculated as follow:

$precision = \frac{8}{8+2} = 0.8 = 80\%$

$recall = \frac{8}{8+3} = 0.72 = 72\%$

$F = 2 \times \frac{80 \times 72}{80+72} = 75$

Performance

The CUSUM approach is implemented on the Circus Oz data set to detect applause sound. This implementation uses the same experiment parameters as those proposed by Sarala et al. [2012]. The parameters are:

- CUSUM value > 0
- applause duration > 0

This experiment is conducted using the evaluation approach described in Section 5.1. The results of the experiment are shown in Table 5.2 where the average recall is 85.87%, the average precision is 25.42% and the F value is 39.22. The precision value is low because there are too many false positives. The CUSUM technique quite often detected a non-applause clip, mostly music, as an applause clip. Furthermore, some of the applause could not be

audio_id	recall	precision
01_devl	85.62	19.92
03_devl	81.67	21.65
05_devl	93.79	32.36
07_devl	90.34	29.5
09_devl	87.30	21.04
011_devl	76.49	25.42
average	85.87	25.42

Table 5.1: Performance of CUSUM technique on timing of applause evaluation.

audio_id	less_clap recall	less_clap precision	more_clap recall	more_clap precision	pure_clap recall	pure_clap precision
01_devl	76.80	33.60	86.88	39.57	97.54	26.83
03_devl	66.34	41.62	96.19	30.76	99.63	27.61
05_devl	76.79	13.56	96.39	50.58	98.2	35.86
07_devl	86.7	38.34	92.07	53.49	96.49	08.17
09_devl	77.03	39.18	93.50	37.92	99.01	22.91
011_devl	68.40	36.11	76.73	42.97	95.31	20.91
average	75.37	33.74	90.29	42.55	97.71	23.71

Table 5.2: Performance of CUSUM technique on timing of applause evaluation.

detected as either the applause sound is not quite clear or it is confused with another sound (music).

In order to find out which types of applause are detected the most, we mapped the detected applause with the applause classes in the ground truth data. As we can see from the table below, mostly more clap and pure clap are detected. In fact, the recalls of both classes are 90.29% and 97.71% respectively. However, the CUSUM technique struggles to detect less clap clips. The average recall for less clap is 75.37%, while the average precision of the overall class is still low at 25.42%.

In addition, we would like to know the accuracy of the algorithm in detecting the strong applause sound: pure clap and more clap. In this experiment, we grouped pure clap and more clap classes as cl class while less clap and non-clap as nc class.

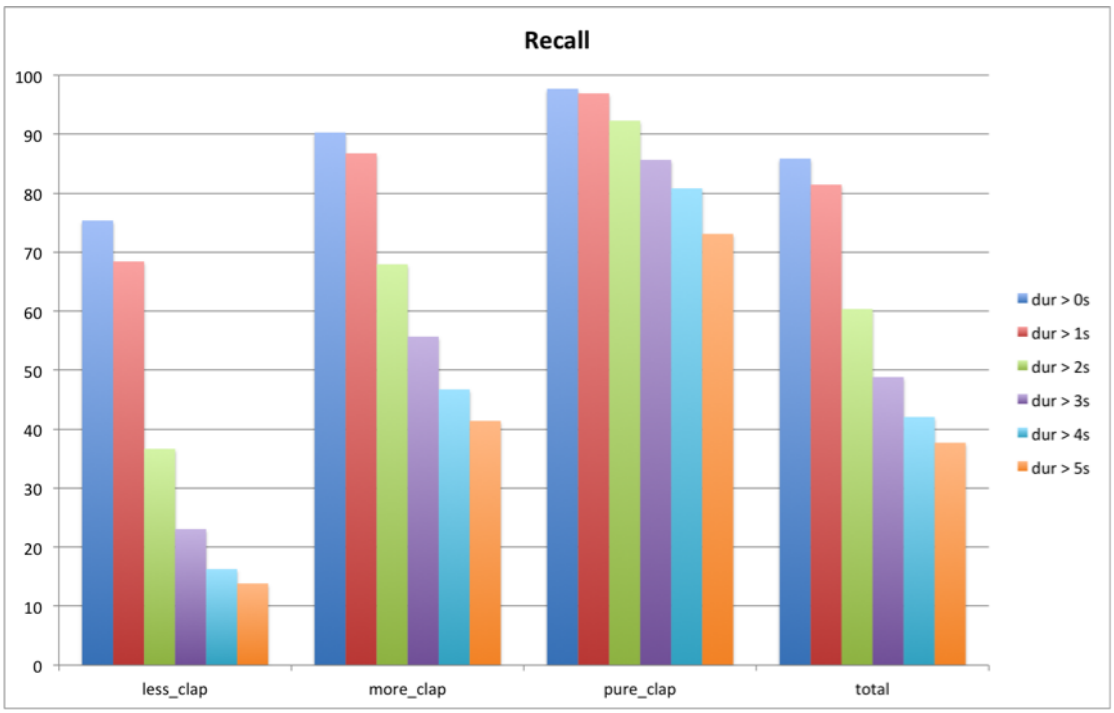


Figure 5.23: Recall of CUSUM technique on different duration threshold

CUSUM technique improvement

In order to improve the recall and precision of the CUSUM technique, we conduct more experiments with a combination of different parameter values.

In the timing of applause evaluation, there are two values that we can evaluate to improve the performance. First is the duration of detected clap. Second is the maximum CUSUM value.

Experiment on applause duration. For the experiment on applause duration, recall, precision and F-value for different applause duration thresholds are shown in Figure 5.23, Figure 5.24, and Figure 5.25 respectively. According to the F-value chart, the optimum minimum duration of the applause sound is between 2 seconds and 4 seconds. Table 5.3 and Table 5.4 show the experiment results for six data sets with an applause duration that is greater than 3 seconds.

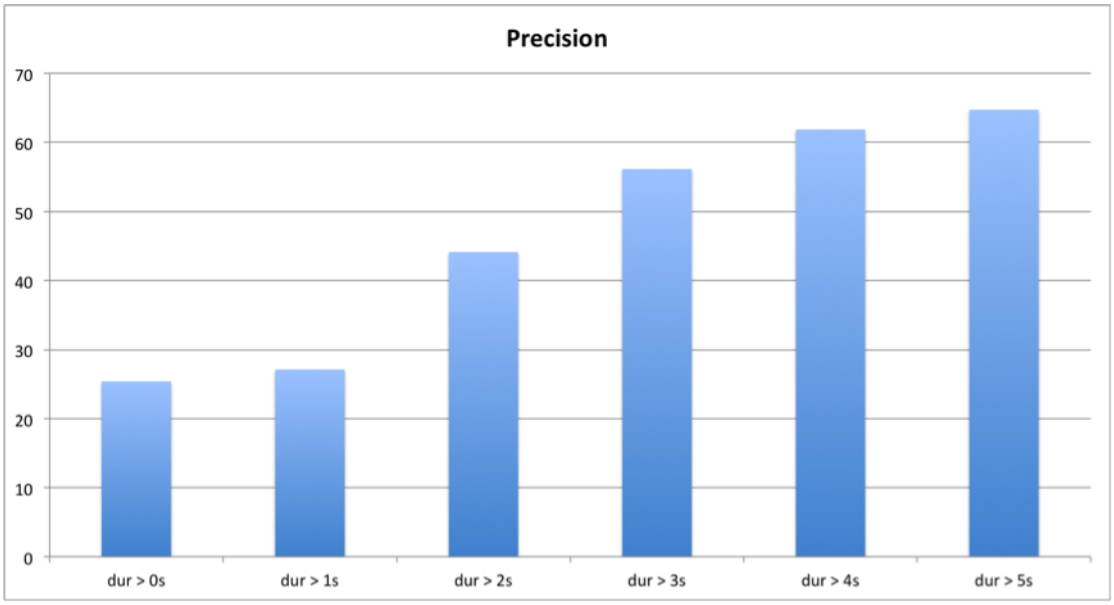


Figure 5.24: Precision of CUSUM technqiue on different duration threshold

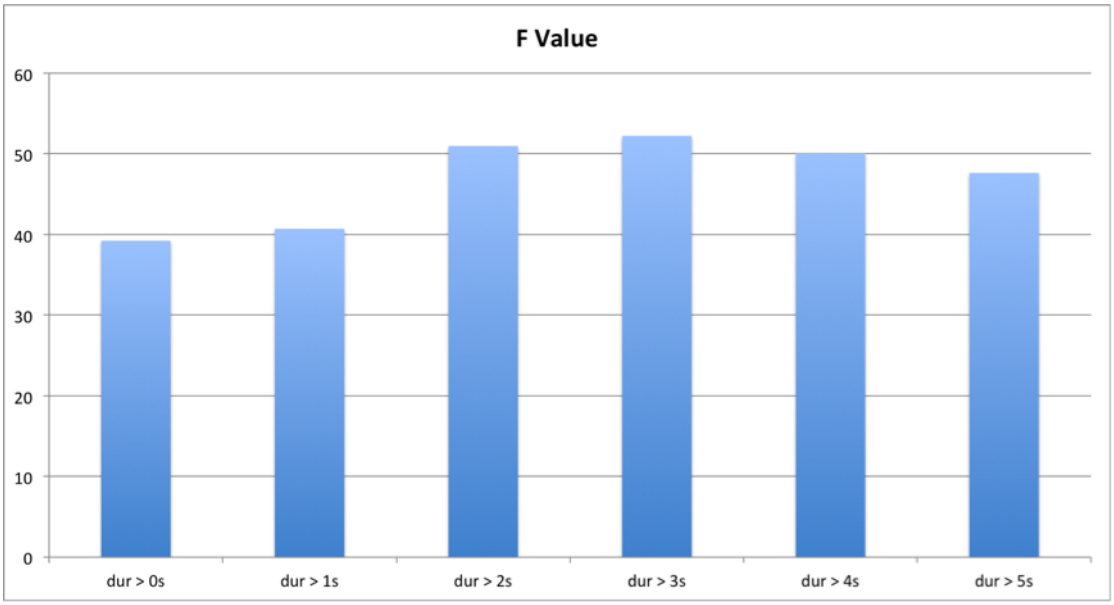


Figure 5.25: F-Value of CUSUM technqiue on different duration threshold

audio_id	recall	precision
01_devl	48.46	40.19
03_devl	45.19	55.73
05_devl	74.06	70.78
07_devl	44.16	73.93
09_devl	54.20	61.04
011_devl	26.92	35.24
average	48.83	56.15

Table 5.3: Recall and Precision applause detection with minimum duration threshold >3.

audio_id	less_clap	more_clap	pure_clap
01_devl	17.78	55.20	86.07
03_devl	15.05	67.94	87.18
05_devl	40.48	69.54	96.83
07_devl	22.56	54.99	82.46
09_devl	30.86	61.30	93.07
011_devl	11.69	25.10	68.23
average	23.07	55.68	85.64

Table 5.4: Recall applause detection with minimum duration threshold >3

Experiment on miniumum CUSUM value. The recall and precision results of this experiment are shown in Figure 5.26 and Figure 5.27 respectively. According to the F-value chart in Figure 5.28, the optimum minimum CUSUM value is 4.

Experiment on both applause duration and CUSUM value. In this experiment, we consider two parameters: applause duration and CUSUM value.

We simply divide the applause classes into two classes: non-clap and clap classes. In the

audio_id	recall	precision
01_devl	56.08	36.89
03_devl	50.91	53.25
05_devl	76.23	76.84
07_devl	49.60	74.05
09_devl	56.30	57.22
011_devl	24.48	32.90
average	52.27	55.19

Table 5.5: Recall and Precision applause detection with minimum CUSUM value >4.

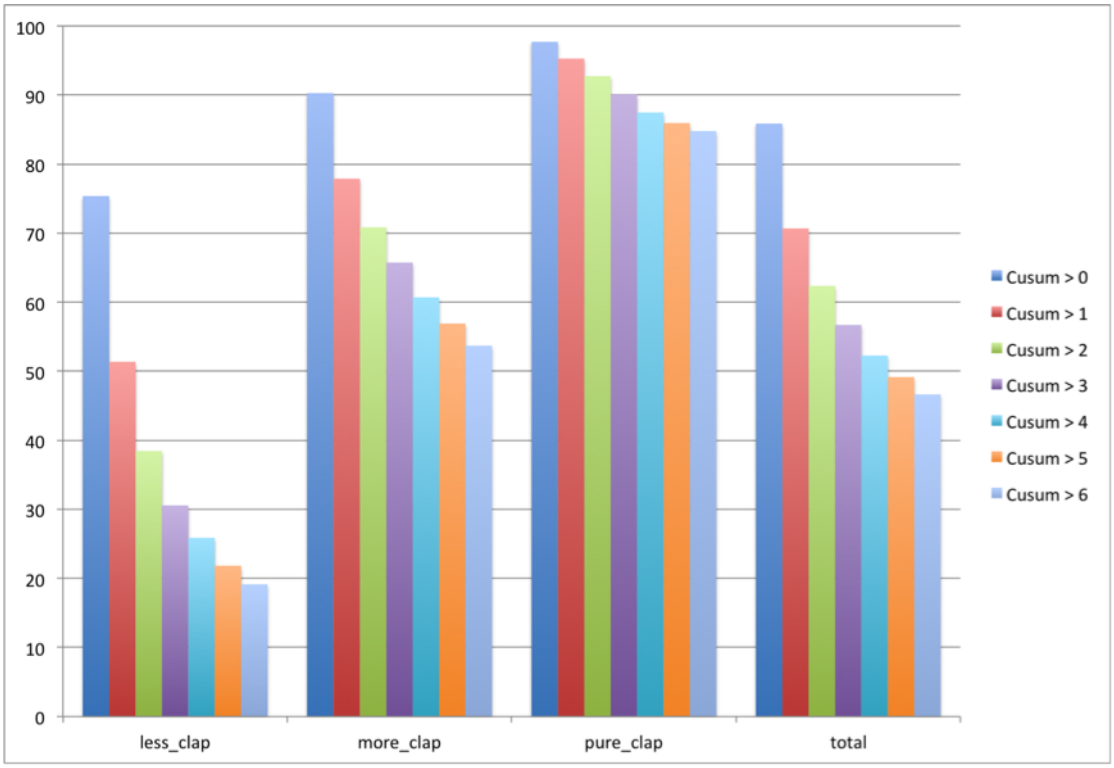


Figure 5.26: The recall of CUSUM technqiue on timing evaluation of various CUSUM values

audio_id	less_clap	more_clap	pure_clap
01_devl	27.58	63.37	89.34
03_devl	19.42	76.19	93.04
05_devl	41.67	73.15	97.41
07_devl	26.77	61.64	85.96
09_devl	30.63	66.38	95.05
011_devl	9.09	23.47	64.06
average	25.86	60.70	87.48

Table 5.6: Recall applause detection with minimum CUSUM value threshold >4

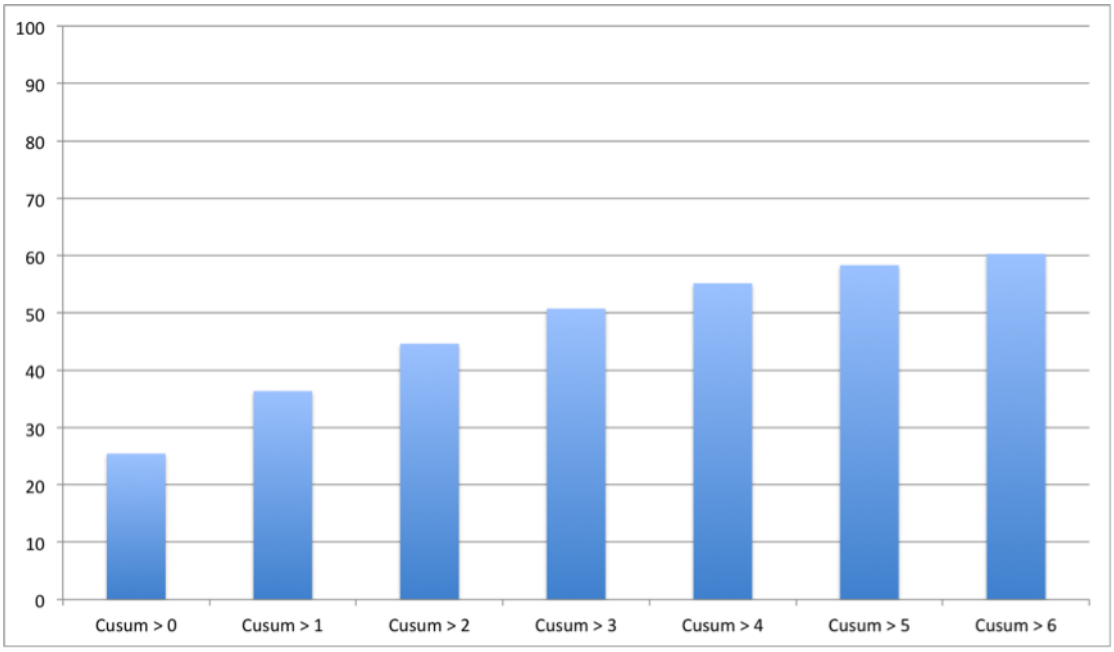


Figure 5.27: The precision of CUSUM technqiue on timing evaluation of various CUSUM values

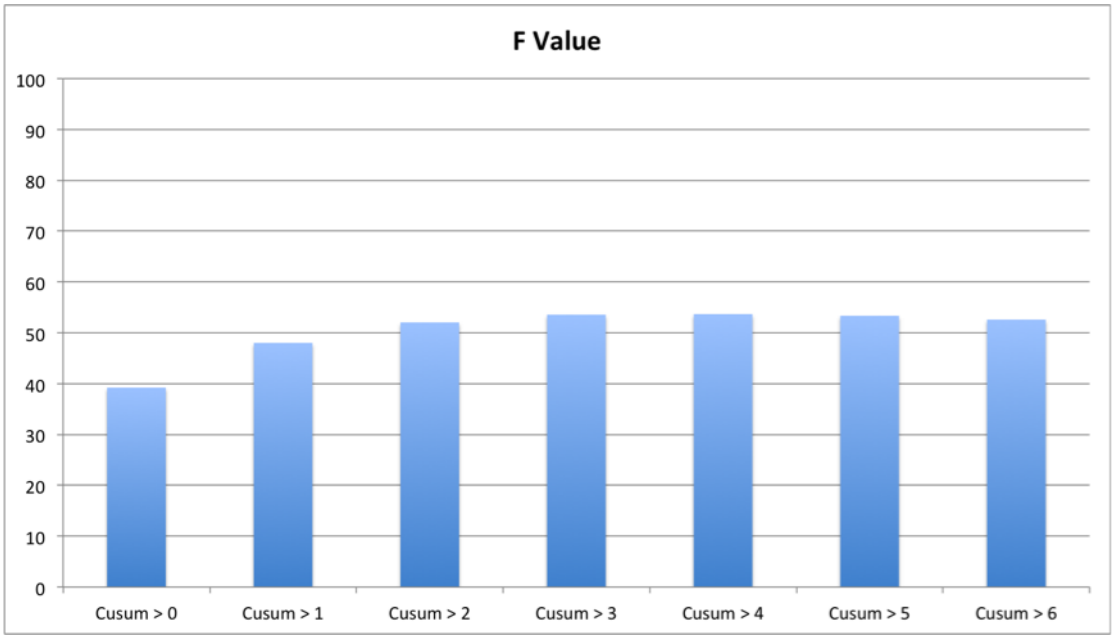


Figure 5.28: The F-Value of CUSUM technqiue on timing evaluation of various CUSUM values

CUSUM / Duration	0	0.5	1	1.5	2	2.5	3
0	38.90	41.42	48.15	51.28	52.56	51.95	50.84
1	47.65	47.67	48.77	51.28	52.56	51.95	50.84
2	51.68	51.68	51.62	51.86	52.63	51.95	50.84
3	53.25	53.25	53.26	52.91	52.77	52.08	50.84
4	53.26	53.36	53.36	53.13	52.56	52.03	50.94
5	53.01	53.01	53.01	52.97	52.72	52.07	51.09
6	52.25	52.25	52.25	52.25	52.12	51.72	51.02
7	51.42	51.42	51.42	51.42	51.42	51.33	50.98
8	50.31	50.31	50.31	50.31	50.31	50.26	49.97
9	49.58	49.58	49.58	49.58	49.58	49.54	49.37
10	48.60	48.60	48.60	48.60	48.60	48.60	48.50

Table 5.7: F-Value applause detection with different CUSUM value and duration threshold.

CUSUM / Duration	0	0.5	1	1.5	2	2.5	3
0	28.64	31.82	41.41	48.31	52.91	54.48	55.65
1	40.01	40.03	42.33	48.31	52.91	54.48	55.65
2	47.56	47.56	47.63	49.39	53.00	54.48	55.65
3	52.30	52.30	52.32	52.29	53.47	54.67	55.65
4	54.79	54.79	54.79	54.69	54.54	55.03	55.77
5	56.18	56.18	56.18	56.18	55.95	55.88	56.09
6	56.81	56.81	56.81	56.81	56.66	56.38	56.38
7	56.99	56.99	56.99	56.99	56.99	56.87	56.81
8	57.03	57.03	57.03	57.03	57.03	56.97	56.86
9	56.48	56.48	56.48	56.48	56.48	56.43	56.41
10	55.96	55.96	55.96	55.96	55.96	55.96	55.86

Table 5.8: F-Value applause detection with different CUSUM value and duration threshold.

first experiment, the non-clap consists of the non-clap class while the clap class includes less clap, more clap, and pure clap classes. The experiment results are shown in Table 5.7. The F-value reached its optimum value when the minimum CUSUM value > 4 and minimum duration > 2 seconds, that is 52.56. This result is better than the F-Value derived from the original CUSUM technique where both the minimum CUSUM value and applause duration > 0, that is 39.22.

In the second experiment, the non-clap consists of non-clap and less clap classes while the clap class contains more clap and pure clap classes. The experiment results are shown in Table 5.8.

5.2 Classification-based approach

5.2.1 Method

We apply machine learning to detect applause where the clips are classified according to particular applause classes based on a given training set. There are two machine-learning approaches: supervised and un-supervised. In the un-supervised approach, the data do not need to be labeled. The data will be classified into a number of clusters. However, in the supervised approach, the data need to be labeled before being submitted to the machine learning algorithm. We focused on supervised data mining to detect applause sound.

There are three steps in the sound recognition analysis. First, selected audio features are extracted such as: beat, MFCC, and spectrum. Second, a machine-learning classifier is employed to build the classification model. The sample sound clips are selected from developed applause data set and these sound clips are then labeled manually according to four classes: non-clap, less clap, more clap, and pure clap. Non-applause sounds comprise music, speech, silence and laughter clips. Then, the labeled audio features are submitted to machine learning software as a training set in order to build a classification model. There are a number of classification algorithms that can be used for sound recognition tasks, such as Bayes Network, Multi Layer Perceptron and Support Vector Machine (SVM). The third and final step is the sound detection task. The audio stream of a Circus Oz video is segmented into a fixed-length clip. Then the selected audio features from each clip are extracted and the machine learning classification algorithm decides the class of each clip (whether clap or non-clap), based on the classification model.

The technique for applause detection will be explained in more detail below and includes audio format, frame and window size for audio feature extraction, suitable audio features for applause detection, and strategy for detecting applause sound.

First is audio format. Most of the audio format for applause detection is 16 bit per sample. This 16-bit audio format is used in experiments conducted by Cai et al. [2003], Olajec et al. [2006], Li et al. [2009] and Shi et al. [2011] experiments. Because their experiments show good results, we use a 16-bit audio format in our experiment.

Second is audio frame and window size. The sizes of frame and window are important for the feature extraction proses. They range from 20ms to 32ms with or without overlapping. For example, Olajec et al. [2006] and Uhle [2011] extract audio features every 23ms with 50% overlap while Cai et al. [2003] and Lesser and Ellis [2005] base their experiments on 25ms with 50% overlap. In our experiment, 23 ms with 50% overlap are used.

The size of clips also varies from one to 30 seconds, depending on the data set being used. For example, Li et al. [2009] split the audio into silence and non-silence and used that as a clip. Another example is the Uhle [2011] data set that ranges from 9 seconds to 30 seconds. Our data set, as explained previously, is split into applause clips and non-applause clips. Non-applause clips are longer than applause clips. Non-applause clips usually last for several minutes, while applause clips usually last for several seconds.

Next is audio features. The selection of an audio feature plays an important role in the classification process. A good feature is one that can distinguish between an applause and a non-applause clip. However, it is not always the case that more features produce a better result. We have to select features carefully based on the characteristics of applause sound. MFCC are the most used features for applause sound detection. Other audio features such as spectral, energy, and PLP are also used for applause detection. In our experiment, we used the following audio features:

- Energy: RMS energy, log energy, and ZCR
- Spectral: Spectral rolloff, spectral flux, spectral centroid, spectral max, spectral min and spectral entropy
- MFCC : MFCC, Delta MFCC, Delta-Delta MFCC
- PLP : PLP, Delta PLP, Delta-Delta PLP

The audio features above are then smoothed and normalized before being submitted to the machine learning algorithm. We used an FIR (Finite Impulse Response) digital filter to

smooth the audio signal as follows:

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j] \tag{5.6}$$

where M = filter size, in our experiment filter size = 10 points.

After that, we normalized the feature value with the following mean, variance, and normalization formulas:

$$m_d = \frac{1}{N} \sum_{n=1}^{N} x_{nd} \tag{5.7}$$

$$\sigma = \frac{1}{N-1} \sum_{n=1}^{N} (x_{nd} - m_d)^2 \tag{5.8}$$

$$\hat{x} = \frac{x_{nd} - m_{nd}}{\sigma_d} \tag{5.9}$$

where N = available data points points of the dth feature ($d = 1, 2, ..., D$).

Finally, the mean of each normalized feature value is calculated using the mean formula given above. The normalized feature value is then submitted to machine learning classifiers.

5.2.2 Performance and evaluation

In this experiment, we use the applause detection technique based on the classification approach. We used the Circus Oz audio development data set as explained in the Data Set Section above with four applause classes: non-clap, less clap, more clap and pure clap.

The audio features are extracted every 23ms with overlapping 50%. The audio sample size is 44.100 samples per second. The audio features included in this experiment are as follows:

- Energy: RMS energy, log energy, and ZCR

- Spectral: Spectral rolloff, spectral flux, spectral centroid, spectral max, spectral min and spectral entropy
- MFCC: MFCC, Delta MFCC, Delta-Delta MFCC
- PLP: PLP, Delta PLP, Delta-Delta PLP

The extracted audio features are then filtered and normalized. The filter uses FIR and normalization is used. After that, we calculate the mean of each small segment of each feature.

We used Weka machine learning software version 3.6.8 to run experiments using various classification algorithms including: BayesNet, Multilayer Perceptron, and Decision Tree. We applied default Weka parameters and settings as they perform well. The classification is run with a 10-fold cross-validation. We measure the performance in terms of the highest number correctly classified.

The first experiment determines the performance of individual audio features. The experiment results (Figure 5.29) show that Spectral, MFCC, and PLP features achieve similar results of approximately 80% being correctly classified using J48 decision tree. However, the energy features achieve a low percentage of correctly classified (72%). In the following experiment, the energy features will be excluded.

Next is the experiment on the derivative features of MFCC and PLP. The experiment results using J48 decision tree (Figure 5.30) show that derivative features are not as good as original features. Hence, in the following experiment, the derivative features are excluded.

Initially, we have tried them all but the performance was not good. The next experiment involves pruning the features. After we have a set of good features, we focus on each subset of features to find which sub set has the most significant contribution to the performance. We used the SFS approach to prune individual features. The energy feature is excluded in this experiment as it obtained the lowest result in the first experiment. In the SFS approach, the evaluation starts with the single features. The best single feature will be included in a 2-features evaluation. The best 2-features evaluation will be included in a 3-features evaluation. This evaluation process will continue if the percentage correctly classified increases and will

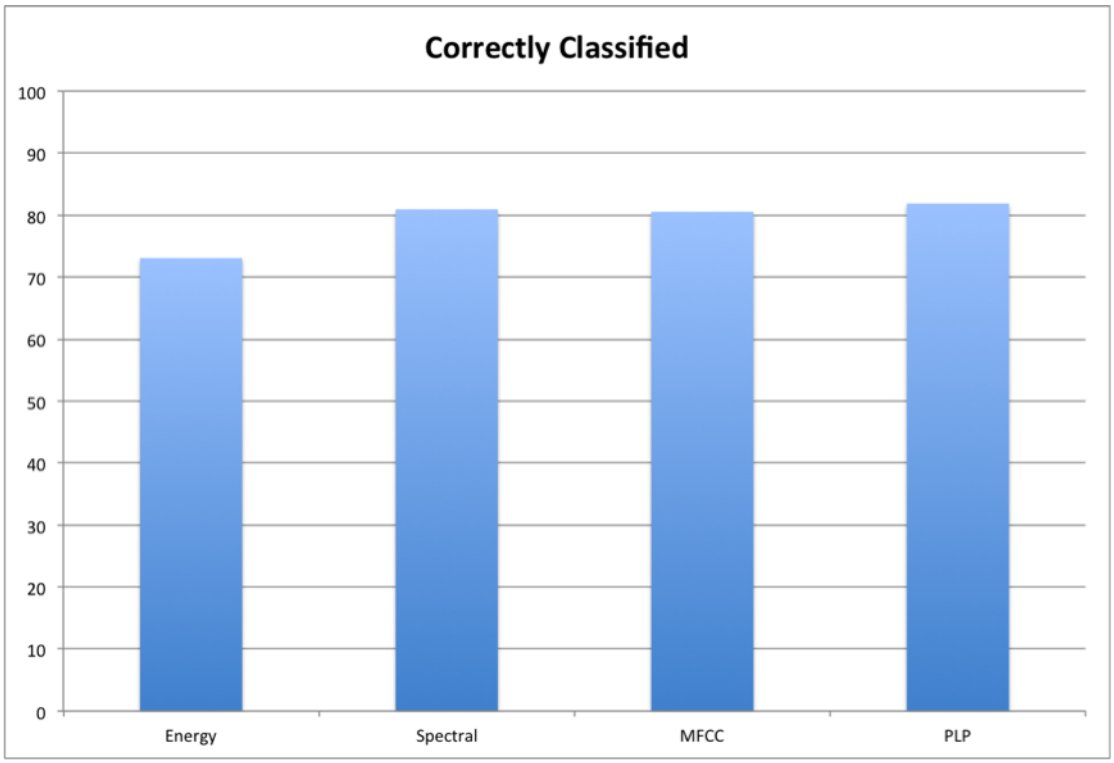


Figure 5.29: Performance of audio features set

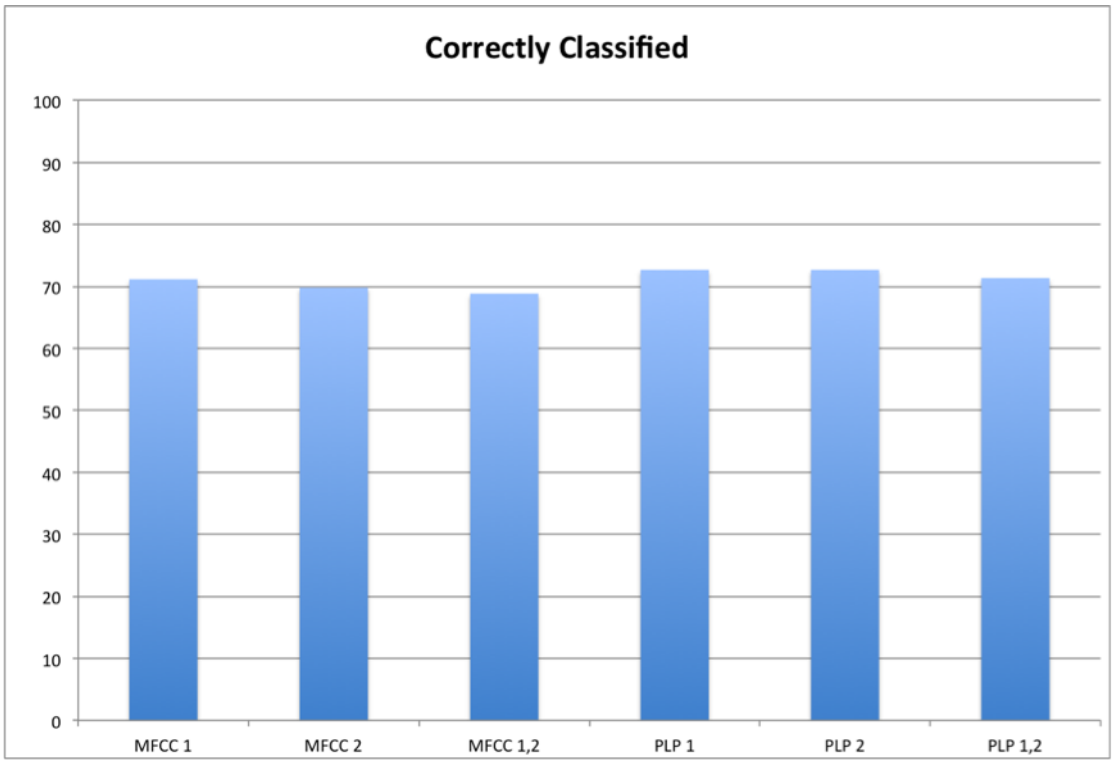


Figure 5.30: Performance of derivative audio features set (MFCC and PLP)

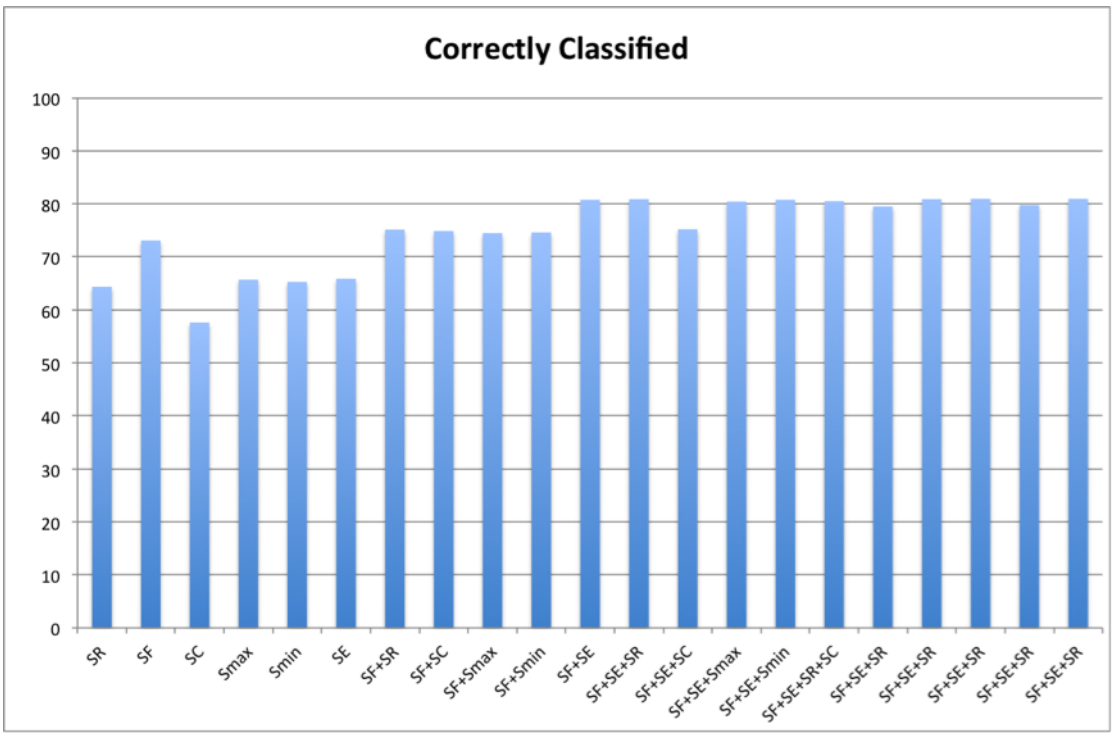


Figure 5.31: Experiment on selecting spectral audio feature set

be stop if the percentage correctly classified decreases.

The results of selecting the best combination on spectral, MFCC, and PLP feature sets are shown in Figure 5.31, Figure 5.32, and Figure 5.33. These experiments are also calculated using J48 decision tree.

The best individual feature sets are:

- Spectral: Spectral RollOff, Spectral Entropy, Spectral Flux, Spectral Min and Spectral Centroid.
- MFCC: 1, 3, 8 and 9
- PLP: all features

The next experiment involves combining different best individual feature sets as follows:

- Spectral SF, SE, SR, Smin, and SC
- MFCC 1,3,8, and 9

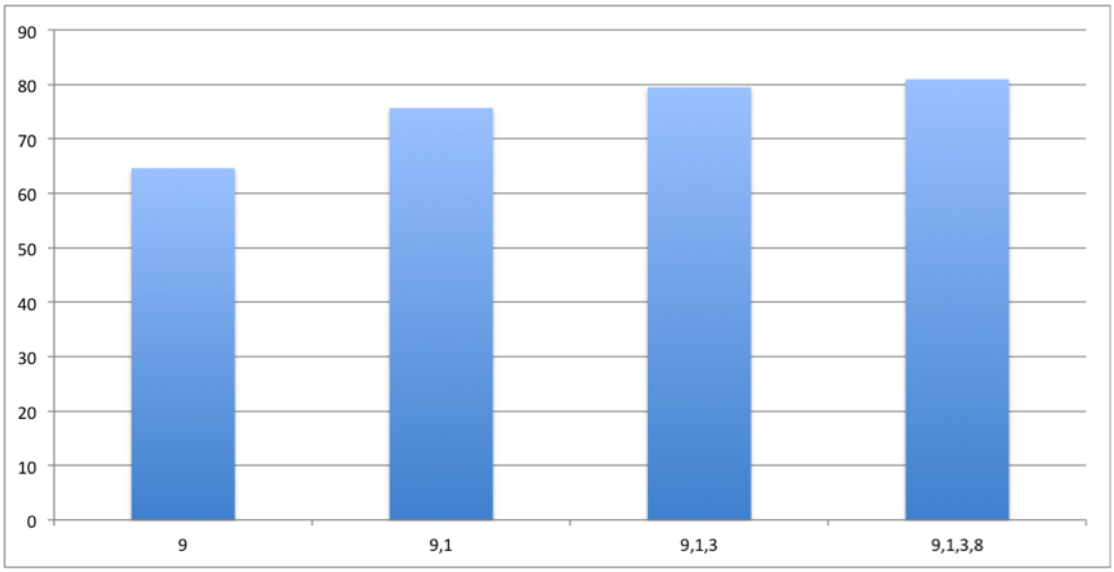


Figure 5.32: Experiment on selecting MFCC audio feature set

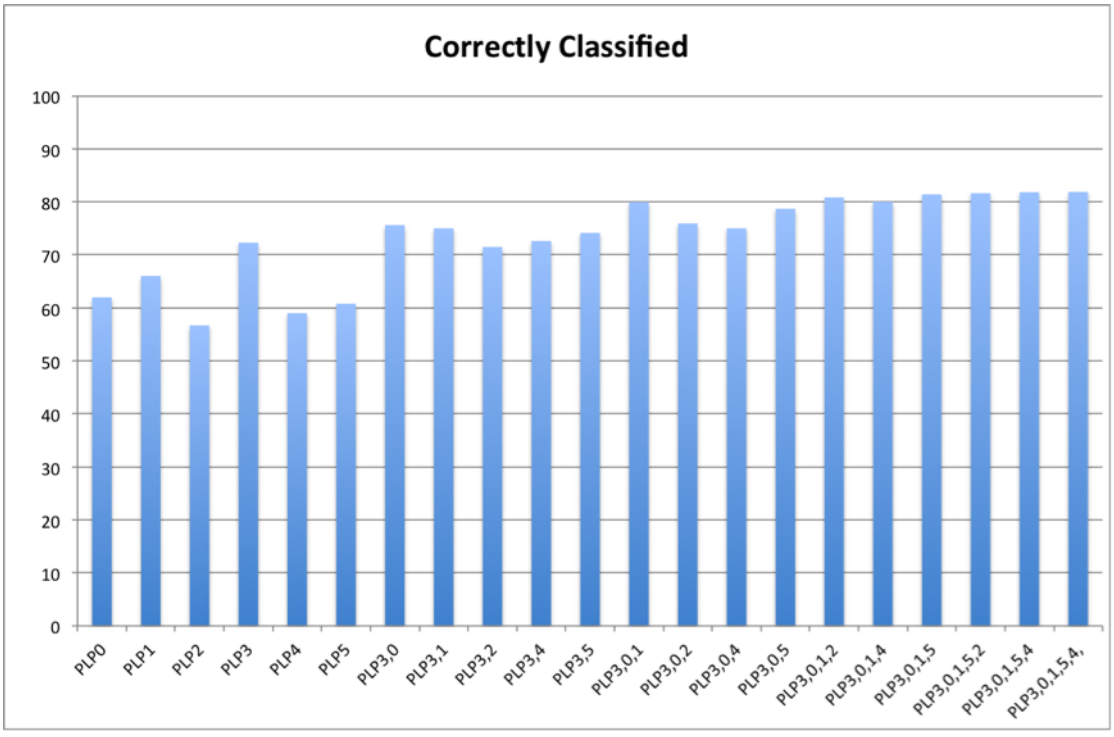


Figure 5.33: Experiment on selecting PLP audio feature set

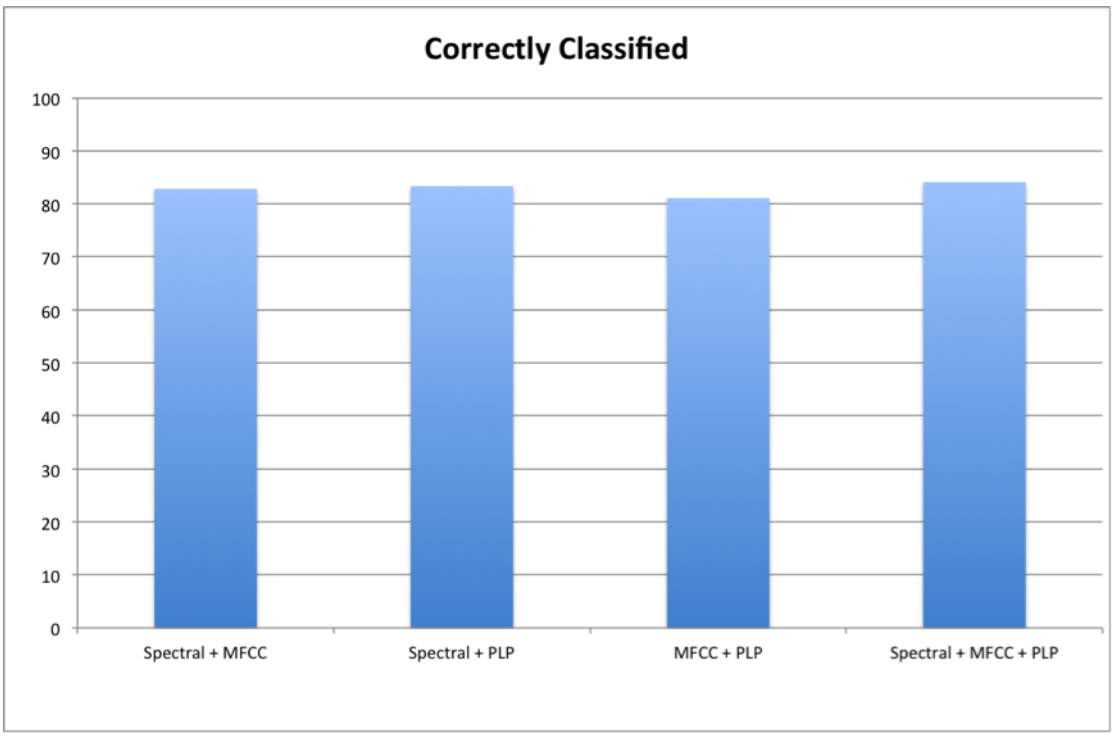


Figure 5.34: Experiment on selecting the best combination audio feature set

- All PLP features

The combination features are:

- Spectral + MFCC
- Spectral + PLP
- MFCC + PLP
- Spectral + MFCC + PLP

The result of the experiment on the selection of the best feature combination set is shown in Figure 5.34.

The combination of features, spectral + MFCC + PLP achieve the best result with 83.33% being classified correctly. The confusion matrix is shown in Table 5.9.

The next experiment involves class mapping with binary and ternary classifications (Table 5.10). Figure 5.35 shows the percentage of correctly classified class on binary and ternary

Classes	non_clap	less	more_clap	pure_clap
non_clap	756	0	0	0
less_clap	0	299	54	4
more_clap	0	108	128	41
pure_clap	0	3	42	77

Table 5.9: Confusion matrix on applause detection with best combination audio feature

ID	Binary Class Mapping		
1	non_clap	less_clap, more_clap, pure_clap	
2	non_clap, less_clap	more_clap, pure_clap	
3	non_clap, less_clap, more_clap	pure_clap	
I	Ternary Class Mapping		
4	non_clap	less_clap, more_clap	pure_clap
5	non_clap, less_clap	more_clap	pure_clap
5	non_clap	less_clap	more_clap, pure_clap

Table 5.10: Binary and ternary classes mapping

class mapping.

The last experiment is conducted to compare classifiers. All classes are used in this experiment: non-clap, less clap, more clap, and pure clap. Four classifiers are compared: BayesNet, Multi-layer Perceptron (MLP), FT Decision Tree, and J48 Decision Tree. As we can see from Figure 5.36, decision tree J48 achieves the best result for every feature.

5.3 Conclusion

In this chapter, we explored the two approaches used to detect applause: characteristic-based and classification-based approaches. First, in the characteristic-based approach, we conducted several experiments applying the CUSUM technique [Sarala et al., 2012] to the circus performance video archive. We found that the optimum parameters when applying this technique are as follows:

- Frame size: 0.01 without overlapping
- Smoothing average filter size: 15 with no symetric calculation
- Audio feature: spectral entropy

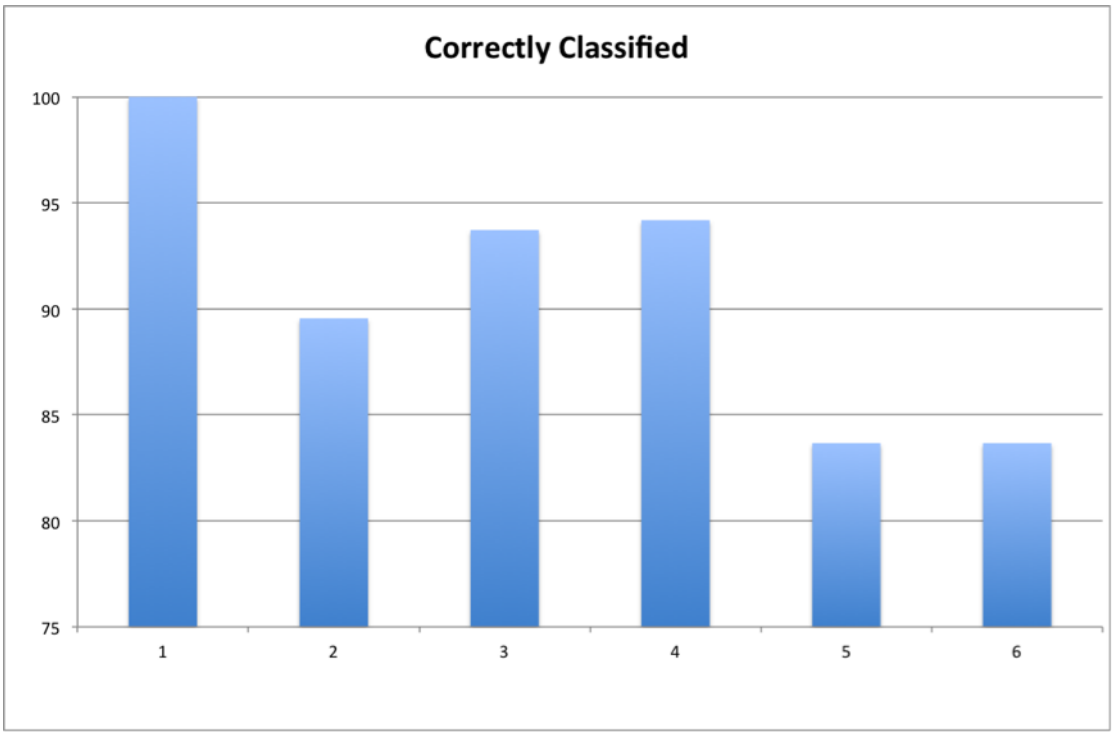


Figure 5.35: Percentage correctly classified applause classes on each binary and ternary class mapping

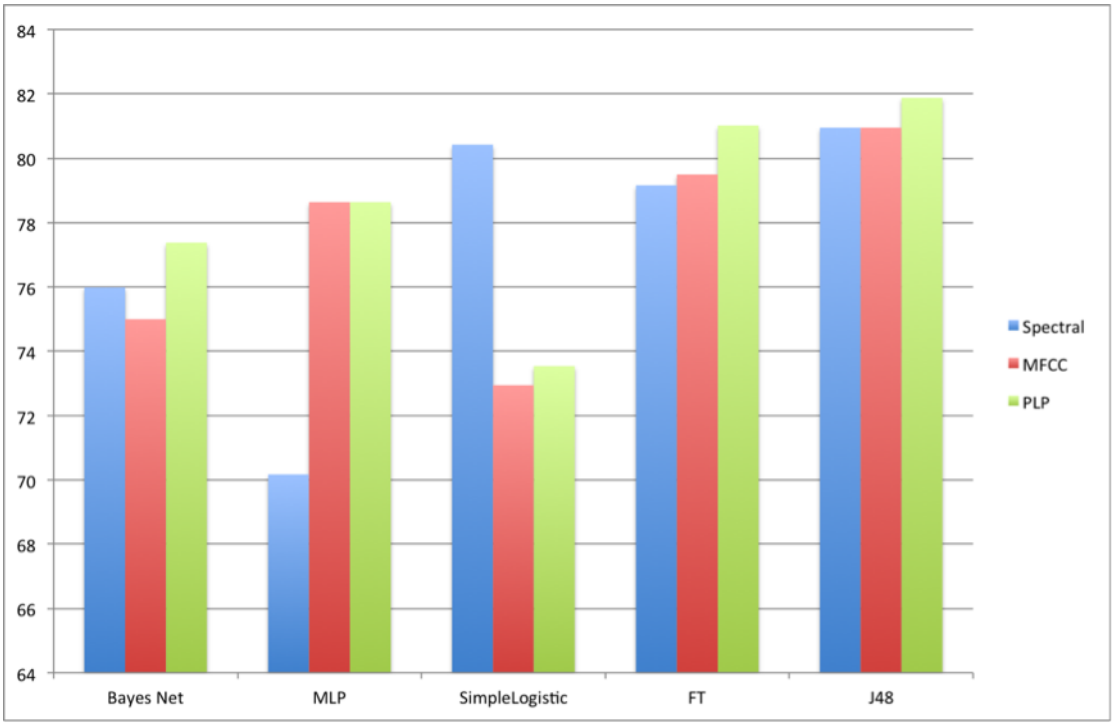


Figure 5.36: Classifiers performance on applause detection

- Audio feature calculation: magnitude spectrum

Furthermore, the minimum CUSUM value and minimum applause duration are important when detecting an applause sound signal. Compared with the original CUSUM technique, the F-value is improved by 13.34% when the minimum CUSUM value and minimum applause duration values are applied.

Second, in the classification-based approach, we used several audio features to detect applause including: Energy, Spectral, MFCC, and PLP. With SFS feature selection, we found that the best combination of audio features is as follows:

- Spectral : Spectral RollOff, Spectral Entropy, Spectral Flux, Spectral Min and Spectral Centroid.
- MFCC : 1, 3, 8 and 9
- PLP : all features

Furthermore, we achieved 83.33% correct classification for the four applause classes. In addition, we mapped the four classes into binary and ternary classes. The best results for the binary and ternary classes are 100% and 94% correctly classified respectively.

Chapter 6

Detecting the End-Of-Act in Circus Performance Videos

In this chapter, we develop a method for segmenting circus performance archival video into acts. Specifically, we investigate how to find a temporal point that is the end time of the act. The general method is described in Section 6.1. This method comprise three parts. They are: applause sound detection, blackframes detection and image comparison. The method and experiment results of each part are explained in Section 6.2, Section 6.3, and Section 6.4 respectively. Performance and evaluation of this method is described in Section 6.5. Finally, Section 6.6 draws the summary of this chapter. The proposed method is implemented on the Circus Oz archive video collection.

6.1 End-of-act detection method

A method for detecting end-of-act clues is shown in Figure 6.1. There are two main steps in this method: applause sound detection and end-of-act detection itself. First, an applause sound detection technique is applied to the circus archive video. The detected applause sound is used as an initial temporal point for end-of-act. A model was built for classifying various sound clips including clap, music-clap, music, laugh, silence and speech clips. We extract various audio features from each sample sound clip. In order to build the model, we apply

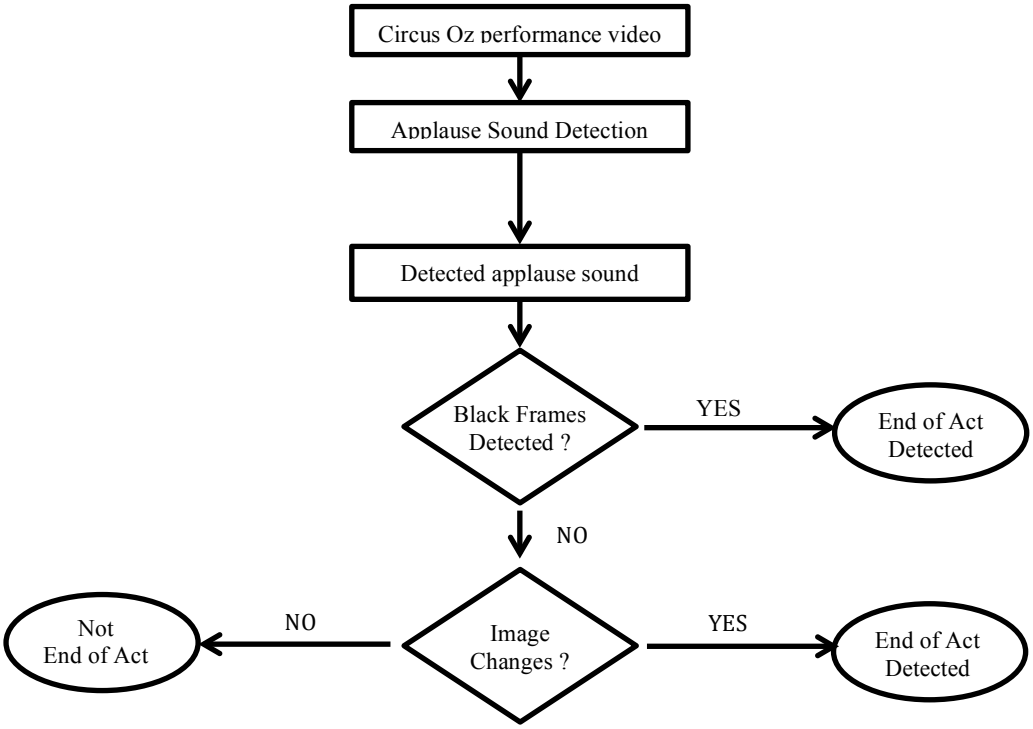


Figure 6.1: End-of-act detection method

several well-known classification algorithms. After that we use this classification model for detecting the applause sound in a circus video. The method and experiment for applause sound detection are explained in Section 6.2.

Second, in order to detect end-of-act, the detected applause sound clips are evaluated as to whether or not the black frames and or image changes occur near the detected applause sound. These clues are used to compare visual content before and after applause sound occurred. The black frame detection technique involves finding the start time and the end time of the appearance (duration) of black frames on the video. To evaluate image changes we used colour histogram comparisons to distinguish the applause sound at the end-of-act from the applause sound in the middle of act. The detailed method for black frame detection and image comparison techniques is explained in Section 6.3 and Section 6.4 respectively.

A strong clue for the end-of-act is when the applause sound is detected and is followed by black frames. Another strong clue for the end-of-act is if the image composition changes. If

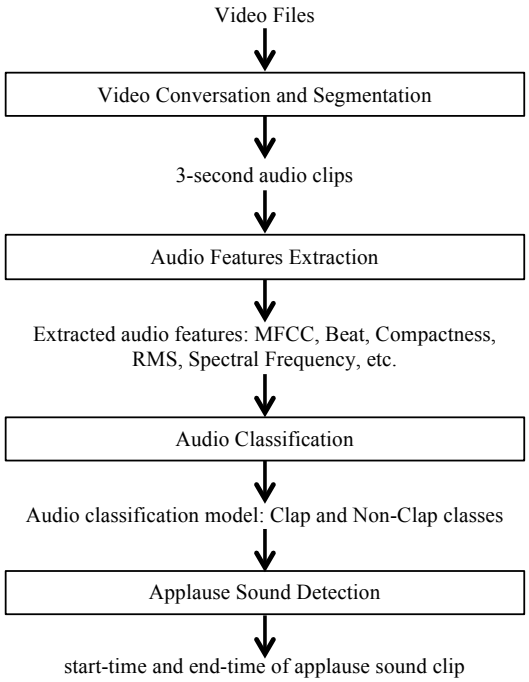


Figure 6.2: Applause sound detection method

neither of these additional features are detected the particular applause sound does not occur at the end-of-act. With the time information that is derived from applause sound detection, black frames and image changes, we then have a strong indication that end-of-act has been detected. The performance and evaluation of this method is described in Section 6.5.

6.2 Applause sound detection

6.2.1 Method

As shown in Figure 6.2, there are four steps involved in the applause sound detection method. In the first step, conversion and segmentation, the original mp4 video file is converted to audio format and segmented into overlapping 3-second wav audio clips. We segment the audio into 3-second audio clips because the typical applause sound duration on the circus videos is at least 3 seconds.

The next step involves audio feature extraction whereby the audio features of each 3-

second clip are extracted using jAudio software [McEnnis et al., 2005]. The aim of this experiment is to detect applause and non-applause sound while the aim of the previous experiment was to detect different applause classes. The extracted audio features are as follows: MFCC, Beat, Compactness, Fraction of Low Energy Frames, Root mean square, Spectral, Frequency and Zero crossings.

In the third step, the above audio features are submitted into Weka data mining software [Hall et al., 2009] to build a classification model based on a selected training set. The 3-second segmented audio file is selected and labeled manually according to one of two classes; clap and non-clap. The clap class comprises pure-clap and mix-clap (clap with music, clap with speech, and clap with cheer) while the non-clap class contains: music, laugh, silence, and, speech. Several well-known classifiers, namely, BayesNet, MLP, SVM-SMO, Decision Tree-J48, FT, and SimpleLogistic are then applied to the training set to build a classification model.

The last step is applause sound detection. As our model segments the audio clip according to a fixed duration of three seconds, we also evaluate each clip according to a three-second window. However, the applause sound might not be detected if its actual starting time is in the middle or at the end of the clip. Moreover, it might not be detected if the actual end time of the clap sound is at the beginning of the clip.

To find the start time and end time of the clap sound, first, we divide the streaming audio source into a number of different segment sets. The duration of each clip for every set is the same, 3 seconds, while each set has a different offset for the start time (and the end time). The offset depends on the degree of time precision that we require. For example, if we want to evaluate the clip at every second, then we can use offsets of 0, 1, and 2. After that, we summarize the results for each second based on the majority rule. If the amount of applause sound is greater than non-clap sounds on a particular 1-second frame, we classify that frame as a clap.

Figure 6.3 illustrates the clap detection process which shows that we have a 9-second audio clip from frame 0 to 8. In the ground truth, each second is labeled manually as c=clap or n=non-clap. The clap sound occurs from frame 2 to 5 on that clip. We would like to

Classifiers	%Correct
BayesNet	87.11
Mulilayer Perceptron	96.77
SVM-SMO	95.11
J48	89.88
FT	93.55

Table 6.1: The Accuracy of 2-Class Classification.

evaluate every 1-second clip. As our model is based on a 3-second clip and we would like to evaluate every 1-second clip, first, we divide the source clip into three sets (set 1, set 2 and set 3) with different start times. The start time of each set is 0, 1 and 2 respectively. Therefore, the first segment frames of set 1, set 2 and set 3 are within the ranges of 0-2, 1-3 and 2-4. After that, the classification model predicts each segment frame on set 1, 2 and 3. On set 1, set 2 and set 3, the clap sound is detected in ranges 3-5, 1-6 and 2-4 respectively. None of them is predicted correctly as the ground truth. Last, we summarize the results from each set for each second. For example: the result of frame 2 is clap as there are more sounds of the clap class than non-clap class while the result of frame 6 is non-clap as there are less clap class than non-clap class. Overall, the result is the same as the ground truth where the clap is detected from frame 2 to 5.

6.2.2 Performance and evaluation

In the applause sound detection experiment, we tested the performance of the classification model and the accuracy of applause sound detection. In the audio classification experiment, we chose one video from each year (from 1983 to 2012). We selected a total of 30 videos as a training set. Each video has 30 sample clips divided into 6 classes. They are 5 clap, 5 music + clap, 5 music, 5 laugh, 5 silence, and 5 speech classes. Each clip has a 3-second duration. The total training set is 900 clips of 45 minutes duration.

The overall result of 2-classes classification with 10-fold cross validation is shown in Table 6.1. The multilayer perceptron classifier clearly outperformed other classifiers. In fact, the multilayer classifier achieved 96.77% accuracy in identifying 2 classes: clap and non-clap.

The confusion matrix of the 2-classes classification is shown in Table 6.2. This matrix is

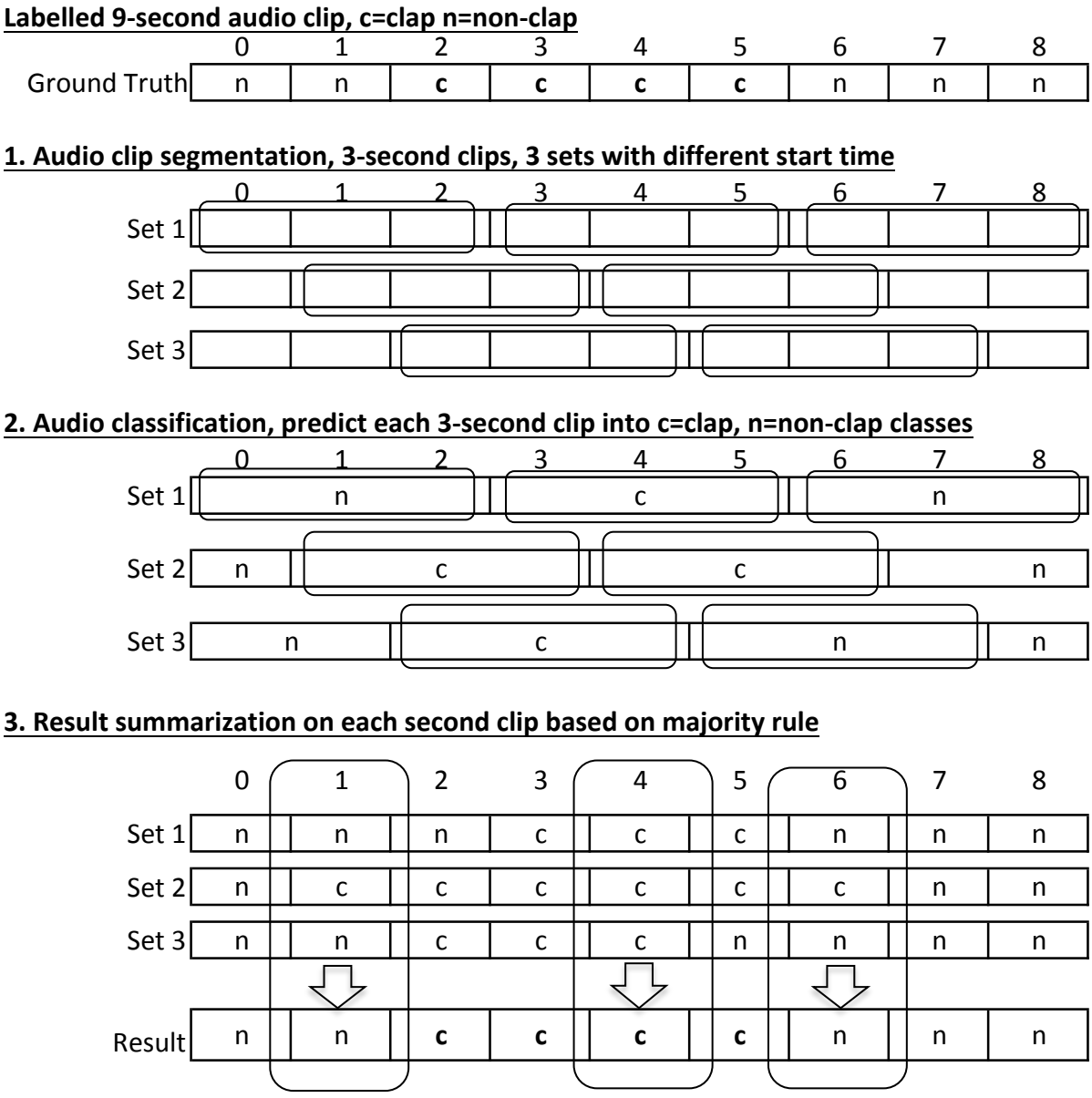


Figure 6.3: Applause sound detection method

classes	clap	non-clap
clap	94.7	5.30
non-clap	2.40	97.60

Table 6.2: The Confusion Matrix of 2-Class Classification.

Data Set	Precision	Recall
Set 1	72.08	86.34
Set 2	71.54	85.86
Set 3	60.84	83.13
3-sets with summarization	73.12	86.84

Table 6.3: The performance of clap detection method

generated using a multilayer perceptron classifier. Both clap and non-clap classes have very good classification results. In fact, the clap class is predicted with 94.7% accuracy while the non-clap class has an accuracy of 97.7%. Thus for the 2-classes classification, we achieved similar as the speech and non-speech classification reported in the literature.

In the applause sound detection experiment, we used 30 stream audio clips as data. These 30 clips were taken from different years, one video per year (from 1983 to 2012). As we are focussed on detecting end-of-act, a clip contains a single circus act from beginning through just before start of the next act such as aerial, juggling, pole, wheels, bikes, acrobalance, and specialist acts. The duration of each clip varies from 2 to 14 minutes. The total duration of the validation set is 02:35:59. Each clip has sub three sets: set 1, set 2, and set 3, with different segmentation. Each set is segmented into 3-second clips. Each set has 3,119 segmented clips. The total number of segmented clips is 9,359.

The performance of the clap detection method on the validation set is shown in Table 6.3. 73.12% precision and 86.84% recall was recorded using multiple subsets. The precision and recall are improved by using multiple subsets compared with a single set method. In fact both precision and recall performances on all single sets (set 1, set 2 and set 3) are lower than the performance of multiple subsets.

Figure 6.4: Act transition with black frames

6.3 Black frames detection

Light intensity changes play an important role not only in circus performances but also in other live performances such as music and dance. In Circus Oz performances the light intensity often changes from high into low intensity or vice versa, and frequently the lights are turned down to low intensity or the light color is changed at the end-of-act. This communicates to the audience that the next act is coming as well as giving time to performers to prepare set changes for the next act without distracting the audience.

When the light is low intensity the visual content of these frames appears darker compared to other frames. Although Circus Oz performances are generally recorded in a low light intensity environment, turning down the light intensity dramatically is visually different, noticeable in video as black frames. Figure 6.4 shows how the black frames appear for a couple of seconds after the roof-walk act has ended and before a group juggling act begins. The black frames can appear in the middle-of-act or at the end-of-act, however these two can usually be distinguished by their duration. Most black frames in the middle of an act are less than 2 seconds long while most black frames at the end of an act are longer than 2 seconds.

6.3.1 Method

The aim of the black frame detection technique is finding the start time and the end time of black frames appearing on a video (that is their duration). We used the ffmpeg multimedia tools to detect these black frames. The video frames were classified as black frames if they met both the following conditions:

1. The minimum duration of the black frames is greater than the *duration threshold*.
2. The ratio of black colour on a frame is greater than the *black ratio threshold*.

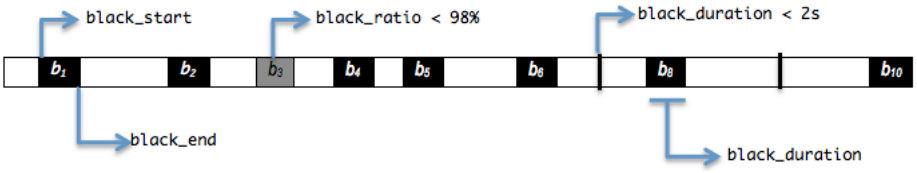


Figure 6.5: Blackframes detection

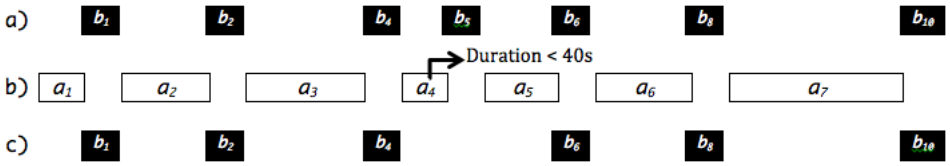


Figure 6.6: Blackframes detection refinement process

In addition, the *colour range* classified as black can be set as a parameter. The output of the black frames detection technique is the start time, the end time and duration of the black frames in the video.

Figure 6.5 illustrates the black frame detection process. On one hand, although the duration of frame b3 is greater than the minimum duration, b3 is not classified as a black frame, as the ratio of black on that frame is less than 98%. On the other hand, the small frame duration between b6 and b8 is not detected as a black frame as the duration of that frame is less than 2 seconds.

However, the black frames do not always or only appear at the end of an act on a circus video. These detected black frames could be related to camera operation where a black spot is shot or the audience may be blocking the camera. Therefore, the detected black frames need to be refined.

The refinement process is shown in Figure 6.6. First, the act clips are generated based on the detected black frames as shown in Figure 5.b. After that, the act clip duration is calculated. If the duration is less than the minimum *threshold for act duration*, the clip is merged with the next clip. The act clip duration a4 is less than the threshold value. The a4 is merged with the next clips: b5 and a5. Figure 5.c shows the result of the black frames refinement process where the black frame b5 is removed.

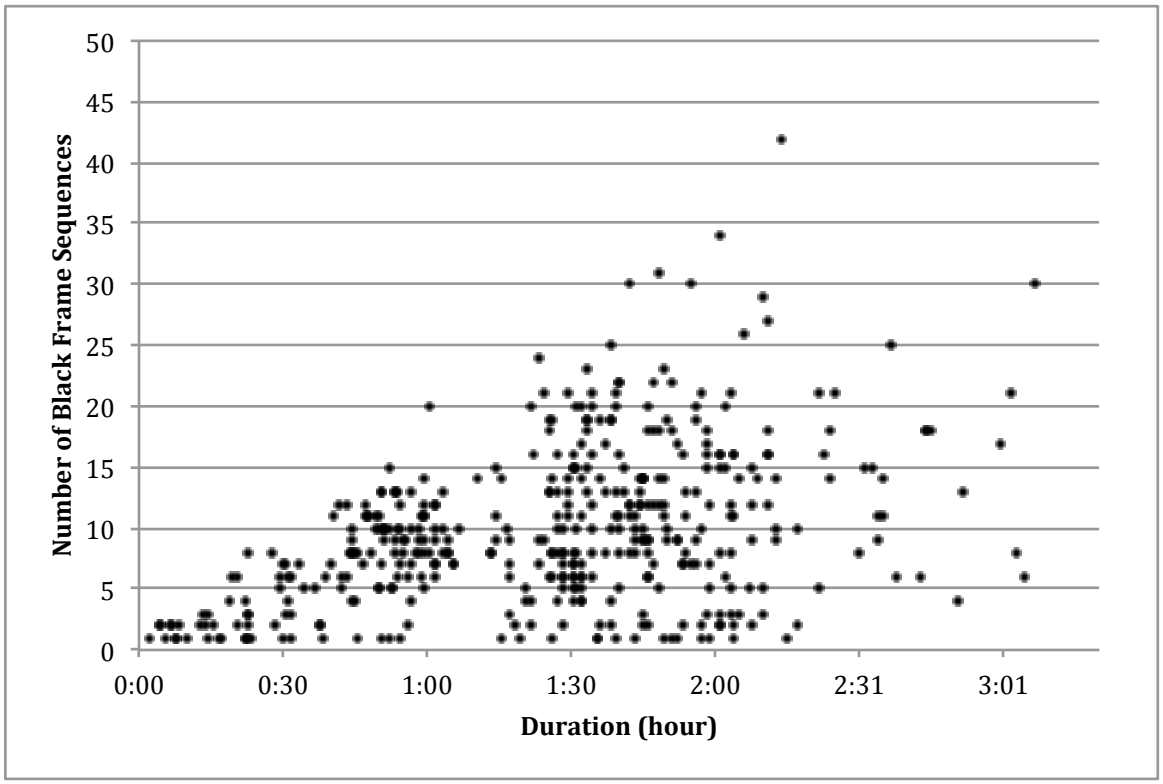


Figure 6.7: The distribution of detected black frames on Circus Oz performance video

6.3.2 Performance and evaluation

In the black frames detection experiment, the value for minimum black duration is set to 2 seconds as the transition time between acts in circus shows is usually at least 2 seconds. As the circus performers present their shows mostly in low light intensity, the black ratio value was set to 98%.

In the implementation we used the following ffmpeg command:

ffmpeg -i inputfile.mp4 -vf blackdetect=d=0.1:pix$_t h = .1 - frawvideo - y/dev/null$

Figure 6.7 shows the result of detected black frames on 420 Circus Oz videos. The dot represents an individual video that has a number of detected black frames for its duration.

6.4 Image comparison

This section explores how to distinguish between an applause sound at the end-of-act and an applause sound in the middle of an act. Specifically, the visual content of the circus video is compared before and after the applause sound occurred in order to find the visual difference between the two.

6.4.1 Method

We use a colour image histogram comparison approach as it is not too sensitive and is less computationally expensive compared to edge detector comparison. The image colour histogram can be compared using two different techniques: image similarity and temporal image frames clustering techniques.

Image similarity

Image content analysis is another way to find that an act has ended in a circus performance video. Two images taken from the same act are likely to have the same image content but two images taken from different acts are likely to have different content (Figure 6.8). The image content across both the backgrounds and foregrounds of images when before and after the end-of-act is compared usually has a noticeably different composition. The background may change from no background (black) to a colorful background affected by light, the number of performers may change from one performer to two or more and the circus apparatus may change from none to the teeterboard, pole, or balancing chairs, for example.

The color histogram comparison approach is used for analyzing image similarity in the Circus Oz videos. This approach is suitable as there is no requirement for exact image matching; rather we are looking for the degree of similarity between the two images. In fact, most images are quite similar overall, as they were taken from one long shot with the same camera angle and position.

There are two steps for comparing image color histograms: histogram calculation and distance measurement. First the color histogram is calculated by counting the number of pixels on each color. The resulting histogram for the examples roofwalk and group juggling

Figure 6.8: Different image content between roofwalk act and group juggling act

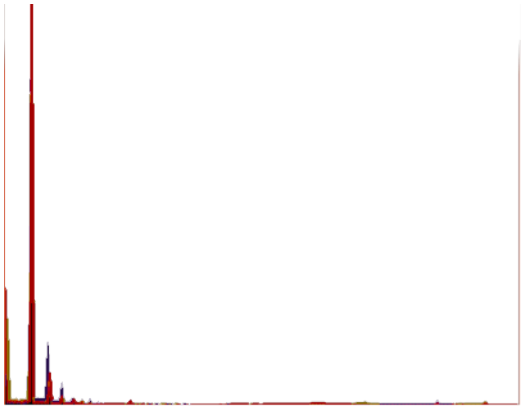


Figure 6.9: Color histogram image taken from roofwalk and group juggling act

images in Figure 6.8 is shown in Figure 6.9. Second, the distance between the two color image histograms is calculated. One way to calculate distance metric of two histogram values is using Correlation formula.

Shot detection

Shot detection software can be employed to detect image changes based on color histogram comparison. The distribution of shot detection analysis on 420 Circus Oz videos is shown in Figure 6.10. Each dot indicates the duration and number of detected image changes in a single video. Apart from act changes, color histogram changes can be caused by many other reasons such as flash light, camera operation and even tape errors such as those caused by converting old analog formats into digital.

When comparing a series of frames from two video clips, there are two options for getting

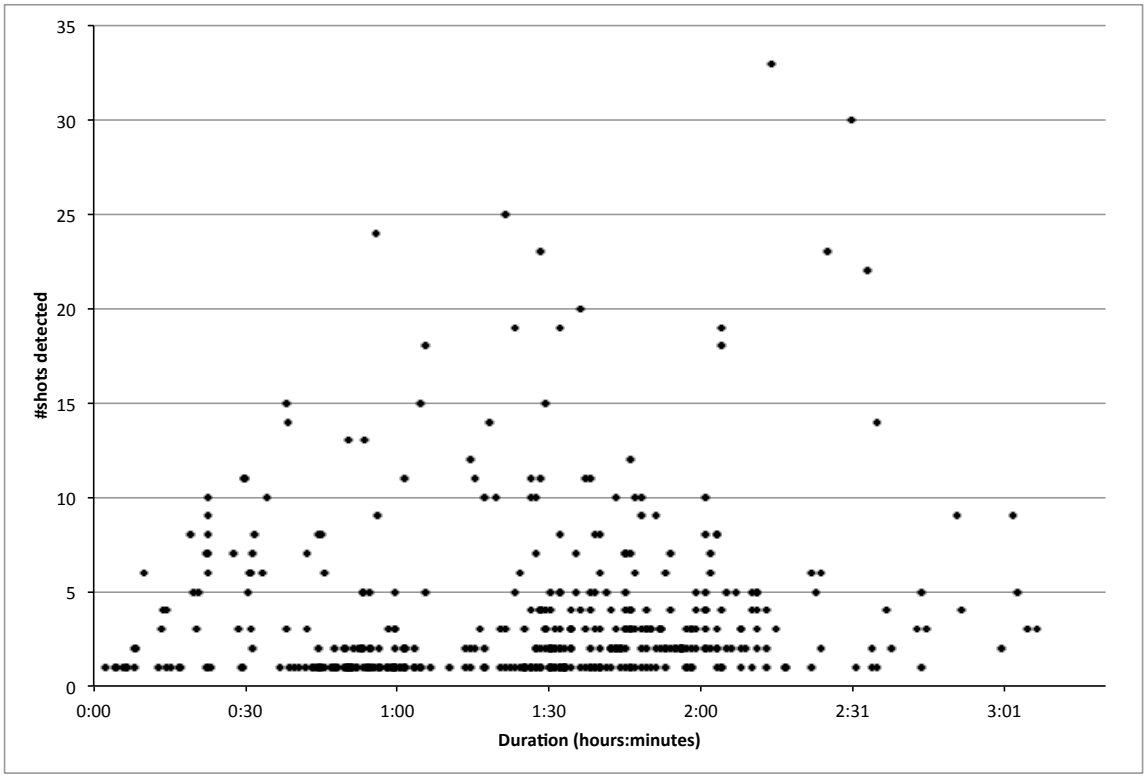


Figure 6.10: The distribution of detected image changes on 420 Circus Oz videos

a similarity Value for these image frames, a) comparing single image frame and b) comparing a series of image frames from each side. To compare single image frames from both sides the steps are as follows:

1) Image Extraction: Two images are extracted: an image frame from n-seconds before the end-of-act and another image frame from n-seconds after the end-of-act.

2) Colour Histogram Generation: The colour image histogram of the two image frames is calculated. Initially, RGB image values are converted into HSV image values. We are only interested in hue and saturation values because they are not too sensitive to image Value changes. Hue value is in the range 0 - 179 (actual hue divided by 2) while saturation value is in the range 0 - 255. We use OpenCV framework version 2.4 for calculating colour histogram that support up to 32 bins of histogram dimension. We segment both hue and saturation ranges into 30 and 32 bins respectively. The colour histogram calculation result is a matrix value with 30 x 32 dimensions.

Hue :$[0, 179] = [0 - 5] \cup [6 - 12] \cup \cdots [174 - 179]$

Hue Range : $bin_1 \cup bin_2 \cup \cdots bin_{30}$

Saturation : $[0, 255] = [0 - 7] \cup [8 - 15] \cup \cdots [248 - 255]$

Saturation Range: $bin_1 \cup bin_2 \cup \cdots bin_{32}$

3) Image Similarity Calculation: The two-image histogram values are then measured for their similarity. We use a correlation formula to measure this.

4) Similarity Threshold Value Definition: Finally, an image similarity threshold value is defined. Two images are defined as similar images if their image similarity value is greater than the defined threshold value. Otherwise, these two images are not similar. The image similarity value is obtained by comparing colour image histogram between two images.

Another option is comparing a series of different two image frames to get a percentage similarity between the two series of frames. The image frames to compare is one series of image frames taken before the applause sound occurred and another series of image frames taken from after the applause sound occurred. The steps for comparing a series of image frames are similar to comparing single image frames, with a difference in Step 1 and Step 2. On Step 1, instead of extracting single image frames, a series of image frames are extracted

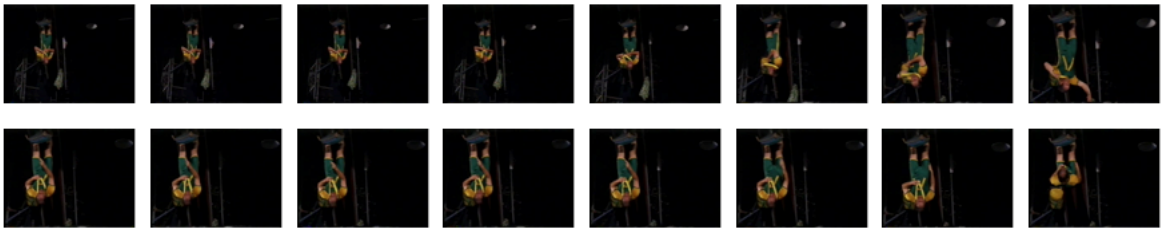


Figure 6.11: Extracted 16 images : 8 images (top row) taken before applause sound and 8 images (bottom row) taken after applause sound in the middle of act

Figure 6.12: Extracted 16 images that 8 images (top row) taken before applause sound and 8 images (bottom row) taken after applause sound at the end of act

between n-second to end-of-act and between end-of-act to n-second. Figure 6.11 shows a series of image frames taken from different two-points in the middle of act while Figure 6.12 shows a series of image frames taken from just before the end-of-act and in the beginning of the next act. On the last step, an average colour histogram value from a series of image frames is calculated instead of a single colour image histogram calculation.

Temporal image frames comparison with clustering

The image similarity technique above requires a similarity threshold value in the last step. However, the threshold value may differ from one video to another as the visual quality of the content varies. In order to solve that issue, we propose a temporal image histogram with a clustering technique. This technique is the same as the above approaches that use colour histogram calculation and compare a series of image frames; however, it is different in terms of measuring the image similarity. Instead of calculating the image similarity, this approach employs a clustering algorithm to cluster a series of image frames before and after an applause sound occurred. Therefore, the similarity image threshold value is no longer

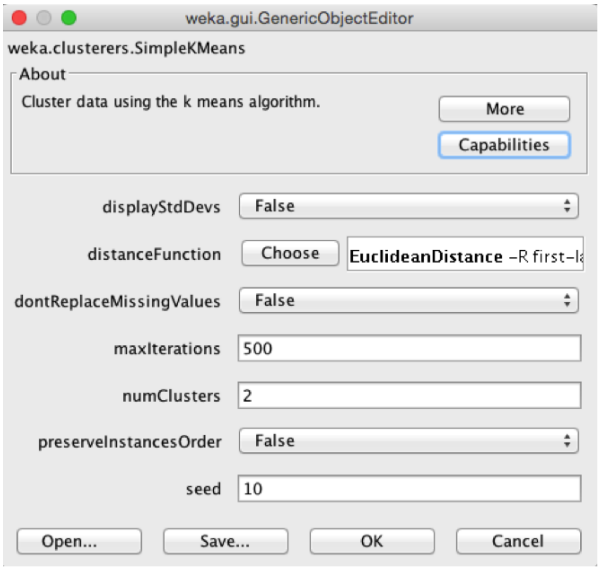


Figure 6.13: Simple K-Means experiment parameters and settings on Weka version 3.6.8

needed.

The steps for comparing a temporal image histogram with a clustering technique is as follows: Step 1 and Step 2 are the same as the image similarity technique above, that is a series of image frames are extracted between n-second to end-of-act and between end-of-act to n-second; and a colour histogram of each image is calculated. In the last step, the colour histogram values are then submitted to a clustering algorithm. The purpose of this is to find out whether or not they are clustered into two clusters or one cluster using clustering algorithms, such as: Simple K-Means [Arthur and Vassilvitskii, 2007], X-Means [Pelleg and Moore, 2000] and Hierarchical Clusterer [Johnson, 1967] algorithms. If they are clustered into two clusters then it is likely that the applause sound occurred at the end-of-act. Otherwise, the applause sound occurred in the middle of the act.

We used Weka machine learning software version 3.6.8 to run experiments. The parameters and settings for Simple K-Means are shown in Figure 6.13.

6.4.2 Performance and evaluation

In order to compare the image before and after an applause sound occurred, we set the following parameters for our experiment: First, hue and saturation image features were used

Data Set	Precision	Recall
1	72.73	47.06
2	100.00	25.97
3	73.33	55.00
4	27.78	10.64
5	18.18	33.33
6	95.83	37.70
7	45.00	37.50
8	92.31	25.53
9	100.00	26.67
10	71.43	53.27
Average	69.66	35.10

Table 6.4: Single image frame histogram comparison result

to generate an image histogram. Second, image resolution was set to the same resolution as a web video resolution that is 320 x 420 pixels. Last, the image was taken 5 seconds before an applause sound and 5 second after an applause sound, the reason being there are too many visual noises between 5 second before and 5 seconds after applause sounds including camera operation (zooming and panning) and flash light noises.

In the implementation, we used OpenCV framework to calculate the histograms and to compare them as the following functions:

calcHist(&bgr_planes[0],1,0,Mat(),b_hist,1,&histSize,histRange,uniform,accumulate);

compareHist(hist_base,hist_base,compare_method);

We conducted three experiments to compare the image comparison techniques above: single image frame similarity, series of image frames similarity and temporal image frames comparison with clustering. In the single image frame similarity experiment, we compared two image frames: an image frame taken from 5 seconds before applause sound and another image frame taken from 5 seconds after applause sound. The image similarity threshold was set to 90%. This experiment uses the method described in Section 6.4.1 The experiment result on 10 Circus Oz videos is shown in Table 6.4.

In the series of image frames similarity experiment we compared two series of image frames: an image frame taken from 5 to 10 seconds before an applause sound and another image frame taken from 5 to 10 seconds after an applause sound. The number of image

Data Set	Precision	Recall
1	86.36	48.72
2	95.00	26.03
3	86.67	53.06
4	77.78	33.33
5	27.27	42.86
6	95.83	35.94
7	65.00	41.94
8	100.00	16.88
9	100.00	26.67
10	80.95	42.50
Average	81.49	36.79

Table 6.5: Average temporal image histogram comparison result.

frames from each side is 8. The image similarity threshold was set to 90%. This experiment uses the method described in Section 6.4.1. The experiment result on 10 Circus Oz videos is shown in Table 6.5.

In the temporal image frames comparison with clustering similarly experiment we compared two series of image frames: image frames taken from 5 to 10 seconds before an applause sound and other image frames taken from 5 to 10 seconds after an applause sound. The number of image frames from each side is 8 images. Three clustering algorithms are employed to cluster these series of image frames including: Simple K-Means, X-Means and Hierarchical Clusterer algorithms. The experiment result is shown in Table 6.6. The hierarchical clusterer algorithm achieved the best result in both recall (86.87%) and precision (45.89%).

6.5 Performance and evaluation

This section describes the performance and evaluation of the proposed end-of-act detection method described in Section 6.1. We implemented the method proposed method in Section 6.2, Section 6.3, and Section 6.4, namely: applause sound detection, black frames detection and image comparison.

In this experiment, we used 10 Circus Oz videos with a total 15 hours duration of viewing. These 10 videos are selected randomly from different years. The experiment result is shown

Data	Simple K-Means		X-Means		Hierarchical Clusterer	
Set	Recall	Precision	Recall	Precision	Recall	Precision
1	81.82	54.55	72.73	36.36	81.82	54.55
2	90.00	39.13	80.00	21.92	100.00	39.13
3	93.33	53.85	76.67	29.49	93.33	53.85
4	88.89	38.10	83.33	17.24	88.89	38.10
5	81.82	60.00	72.73	32.00	81.82	60.00
6	70.83	41.76	62.50	25.86	70.83	41.76
7	85.00	40.78	65.00	25.49	85.00	40.48
8	100.00	30.23	53.85	17.07	100.00	38.24
9	81.25	30.23	68.75	30.56	81.25	48.15
10	85.71	45.00	66.67	25.00	85.71	45.00
Average	85.87	43.30	70.22	26.10	86.87	45.89

Table 6.6: Experiment result on image histogram with clustering technique

in Table 6.7.

As we can see from Table 6.7, the average recall of proposed end-of-act detection methods is very good with 92.27%. This recall is achieved when both black frames and image comparison are used to evaluated the applause sound. The average precision is low because there are quite a lot of noises on visual features including: camera operations, flashlight, and various video content quality.

Using the proposed end-of-act detection method achieved better results compared to black frames detection or image comparison methods. In fact both recall and precision are improved. Compared to black frames detection, the recall of proposed method are improved from 47.93% to 92.27%. The black frames detection method achieved better precision (76.68%) but the recall is too low (47.93%). Both recall and precision of proposed method are improved compared to image comparison technique. The recall of proposed method increased from 86.87% to 92.27% while the precision improved from 45.89% to 49.05%.

Combining these four techniques (applause sound detection, black frames detection, and image comparison techniques) provides suggestions to humans about where acts are. Humans can more easily divide circus show video into acts when provided with a clue about the end-of-act. Figure 6.14 shows an interface for dividing Circus Oz performance video into acts. The vertical white on time line bar indicate the clues for when an act has ended. This

Data	Black Frames		Image Comp.		Black + Image	
Set	Recall	Precision	Recall	Precision	Recall	Precision
1	40.91	100.00	81.82	54.55	95.45	58.33
2	45.00	90.00	100.00	39.13	100.00	41.67
3	46.67	93.33	93.33	53.85	93.33	53.85
4	27.78	71.4	88.89	38.10	88.89	37.21
5	9.09	50.00	81.82	60.00	81.82	56.25
6	58.33	82.35	70.83	41.46	87.50	43.75
7	95.00	63.33	85.00	40.48	100.00	40.48
8	100.00	30.23	53.85	17.07	100.00	38.24
9	43.75	63.64	81.25	48.15	87.50	77.78
10	66.67	73.68	85.71	45.00	95.25	43.48
Average	47.93	76.28	86.87	45.89	92.27	49.05

Table 6.7: Experiment result on end of act detection on circus show video

interface is proving useful for human annotators to cut the circus video into acts.

6.6 Conclusion

In this chapter we have analyzed the audio-visual content of the circus performance video to find useful segmentation clues that indicate an act within a show has ended. Furthermore, we have proposed three approaches to support the segmentation process.

First, the end-of-act detection method is able to find strong clues that an act has ended. The experiment shows that the precision and recall of the end-of-act-detection method is 49.05% and 92.27% respectively.

Next, the proposed applause sound detection method can be used to detect an applause sound on circus video through a learning process. The experiment shows that the precision and recall of the applause sound detection method is 73.12% and 86.84% respectively.

Third, the proposed image frames comparison with clustering technique is useful for distinguishing an applause sound at the end-of-act from one in the middle of an act. In fact, this proposed technique outperformed the image histogram comparison technique. The experiment shows that the precision and recall of the image comparison with clustering technique are improved compared to the image histogram technique.

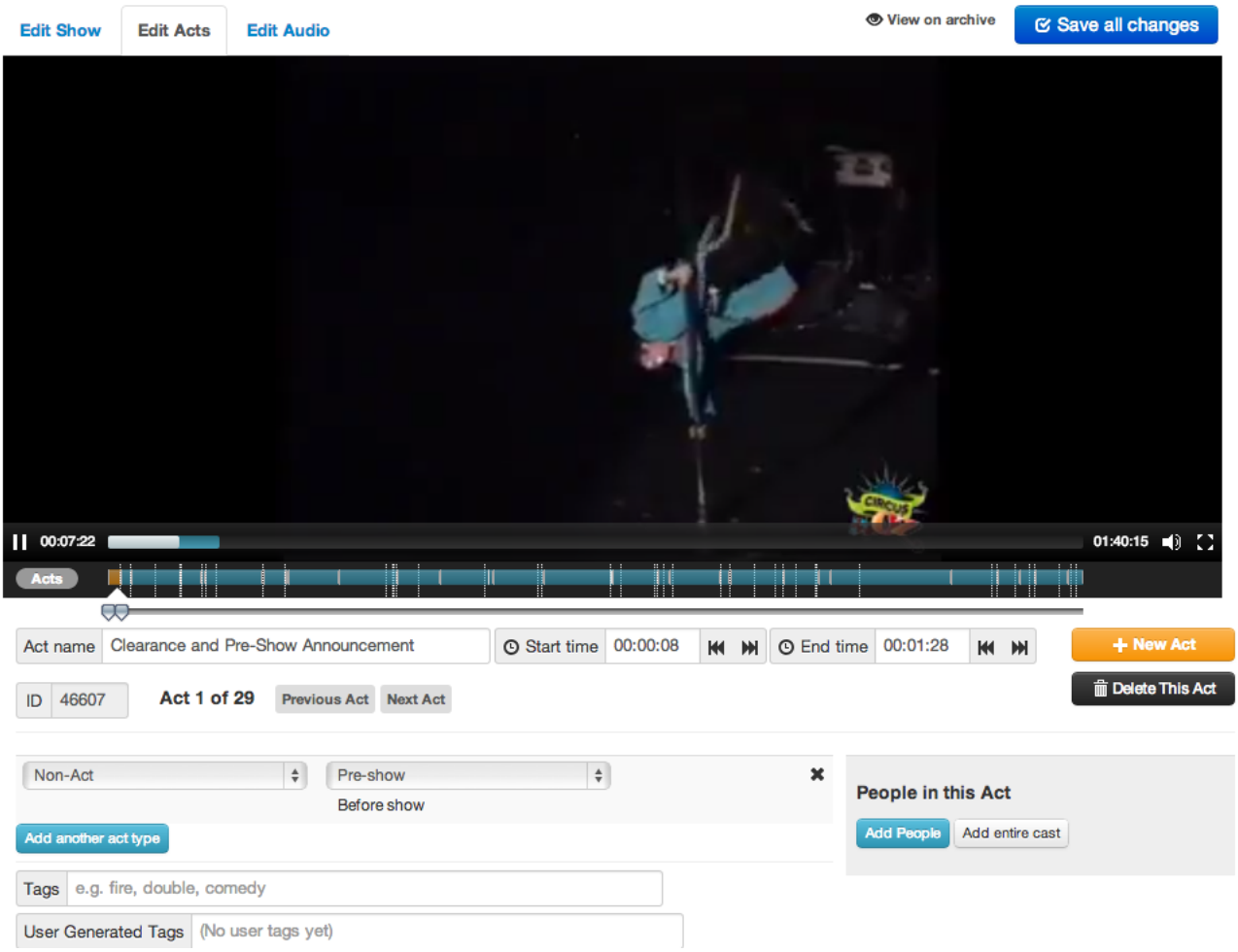


Figure 6.14: Extracted 16 images : interface for dividing Circus Oz show into acts

Chapter 7

Conclusion and Future Work

In this thesis, we have analyzed the audio-visual content of the circus performance video to find useful segmentation clues that indicate an act within a show has ended. We have also proposed methods for detecting applause sound and segmenting long circus video into meaningful clips.

7.1 Conclusion

We have tackled the four research questions of this thesis, that are: developed an effective and efficient performance video archive system for Circus Oz video collection; constructed the applause datasets with three applause classes: less clap, more clap, and pure clap; proposed an applause detection techniques based on characteristic and classification approaches; and proposed a method for automatic video temporal segmentation of circus video into acts.

7.1.1 Circus Oz video retrieval system

A prototype of a video archive system has been developed. The system architecture is efficient enough to handle the unique characteristics of a performance video archive. Although this prototype specifically applies to the Circus Oz performance video archive, it can be used for other performance video archives. The database schema is quite flexible, allowing it to be applied to different video content as long as the system structure is similar in terms of video

and clips relation. The overall system architecture can also be easily implemented in other video retrieval systems. The system architecture consists of five components: video server, front-end application server, database schema, search function, and video processing.

7.1.2 Circus Oz dataset

Applause sound and video segmentation datasets have been developed. For the applause sound dataset, we selected 12 performance Circus Oz videos, all of which are accessible for public access at http://archive.circusoz.com/. The total duration of the entire video data set is 21 hours 19 minutes and 53 seconds of viewing. We have established ground truth in terms of where and when applause sounds occur on the video data set. We defined four classes of applause: non-clap, less clap, more clap, and pure clap.

For the video segmentation dataset, we selected ten performance Circus Oz videos totaling more than 17 hours of viewing. We manually set the end-of-act time ground truth of every act on each video in the dataset. There are a total of 195 acts for all 10 videos. In addition, there are three components of video segmentation dataset: applause, black frames, and images.

7.1.3 Applause detection technique

We have examined the characteristic-based and classification-based approaches to detecting applause. First, in the characteristic-based approach, we conducted several experiments in which we applied the CUSUM technique Sarala et al. [2012] to the circus performance video archive. We found that the optimum parameters when applying this technique are as follows:

- Frame size: 0.01 without overlapping
- Smoothing average filter size: 15 with no symmetric calculation
- Audio feature: spectral entropy
- Audio feature calculation: magnitude spectrum

Furthermore, the minimum CUSUM value and minimum applause duration are important for the detection of an applause sound signal. Compared with the original CUSUM technique,

the F-value is improved by 13.34 when the minimum CUSUM value and minimum applause duration values are applied.

Second, in the classification-based approach, we used several audio features to detect applause. These included: Energy, Spectral, MFCC, and PLP. With SFS feature selection, we found that the best audio feature combination was as follows:

- Spectral: Spectral RollOff, Spectral Entropy, Spectral Flux, Spectral Min and Spectral Centroid.
- MFCC: 1, 3, 8 and 9
- PLP: all features

Furthermore, we achieved 83.33% correctly classified for four applause classes. In addition, we mapped the four classes into binary and ternary classes. The best result for binary and ternary classes is 100% and 94% correctly classified respectively.

7.1.4 Detecting end-of-act in circus perforamce video

We proposed a method for determining temporal act segmentation. First is audio classification and detection. An audio classification model is built using several well-known classifiers such as Multilayer perceptron and SVM-SMO. We achieved an accuracy of 96.77% when distinguishing clapping from non-clapping sounds. We used multiple data sets for detecting clapping sounds on stream circus audio. The experiment shows that the precision and recall of the clap detection method is 73.12% and 86.84% respectively.

Second is black frames detection on the circus video. The black frames detection technique is able to segment a circus video into acts. The application of a minimum act duration threshold reduces the over-detection of segments. In fact, the majority of videos have more than 5 and less than 20 segments. This result is relevant since most of the videos contain between 5 and 20 acts.

Moreover, the temporal video segmentation is done based on the audio-visual content. The combination of the time information of both the detected black frames and the clapping sound provide a strong indication that an act on a circus video has ended.

7.2 Future works

Future work could improve the performance of the applause sound detection technique by combining both characteristic and classification approaches. On one hand, the applause sound detection with characteristic approach is able to detect applause sound efficiently. On the other hand, the applause sound detection technique with classification approach is able to classify multiple applause classes: less clap, more clap and pure clap. Combining both approaches may lead to improve the performance of this technique to detect the applause sound and classify them into different applause sound classes.

Another direction for future work would be aiming to improve the precision of the end-of-act detection method. As there is too much noise in the visual content of the Circus Oz videos, more analysis on audio content would be preferable; that is comparing the music before and after the end-of-act. Audio features in the middle-of-act clip potentially contain similar audio content while the audio features at the-end-of-act is likely to contain different audio content. Furthermore, applying a machine learning approach on the end-of-act detection method may improve the accuracy of the temporal video segmentation technique. Given audio and visual features, we can than process these features using a machine learning algorithm.

Another solution to segment circus video into acts is by analyzing movement in the video. Different acts have different movement pattern in circus video. This can be done by analyzing the image changes from frame to frame. To make the Circus Oz videos collection more usable, the movement pattern of various circus tricks can be developed and published as a data set for further experiments.

Finally, the proposed end-of-act detection method can be applied to segment other video collections that have similar video characteristics and contain black frames and applause sound, such as: music concerts that contain: music, speech, cheering and applause sounds. However, the method may need tuning when applied on other performance video collections. This includes: building the audio classification model for applause detection; black colour ratio and black frames duration thresholds for black frames detection, and the number of extracted images for the image comparison method.

Appendix A

Glossary

ACF Autocorrelation Function

API Application Programming Interface

ARC Australian Research Council

ASR Automatic Speech Recognition

AURORA Automated Restoration of Original Video and Film Archives

BayesNet Bayesian Network

BER Band Energy Ratio

CPU Central Processing Unit

CSV Comma Separated Value

CUSUM Cumulative Sum

FFT Fast Fouier Transform

FIR Finite Impulse Response

FN False Negative

FP False Positive

FT Functional Trees

GA Genetic Algorithm

GB Giga Byte

GHz Giga Hertz

GMM Gaussian Mixture Models

hetMM Heterogeneous Mixture Model

HMM Hidden Markov Model

HR Human Resources

HSV Hue Saturation Value

HTML Hyper Text Markup Language

HTTP Hype Text Transport Protocol

Hz Hertz

IIR Infinite Impulse Response

IP Internet Protocol

Kaltura CE Kaltura Community Edition

KMC Kaltura Management Console

LAMP Linux, Apache, MySQL, and PHP

LLD Low Level Descriptors

LPC Linear Predictive Codes

LSF Line Spectral Frequency

MFCC Mel Frequency Central Coefficients

miniDV Mini Digital Video

MLP Multilayer Perceptron

MP4 MPEG Layer-4

NFS Network File System

OS Operating System

PLP Perceptual Linear Prediction

RAM Random Access Memory

RGB Red Green Blue

RLSC Regularized Least-Squares Classifier

RMS Root Mean Square

SA Simulated Annealing

SC Spectral Centroid

SE Spectral Entropy

SF Spectral Flux

SFS Sequential Forward Selection

Smin Spectral Minimum

SR Spectral Roll Off

STE Short Time Energy

SVM Support Vector Machine

SVM-SMO Support Vector Machine - Sequential Minimal Optimization

TP True Positive

trackThem Tracking Them

TVA TV Appearances

TVC TV Commercial

URL Universal Resource Locator

VAD Voice Activity Detection

VHS Vertical Helical Scan

ZCR Zero Crossing Rate

Appendix B

Video Server

B.1 Kaltura video server

Kaltura has two main sub-systems: Kaltura administration console and Kaltura management console (KMC). The Kaltura administration console is used for administering publishers and monitoring servers and batches. The KMC is used for managing video content and creating video players and publishing.

One example of a main Kaltura video server feature is the bulk video upload. The bulk video upload can be uploaded into a video server. The Kaltura video upload interface is shown in Figure B.1. This kind of upload process is usually needed only at the beginning of project where a lot of video needs to be uploaded quickly. The process of the bulk video upload is as follow: First, all videos have to be accessible to the Kaltura server either through http or ftp protocols. Next, the list of videos and their urls are compiled in CSV file format as shown in Table B.1.Finally, the CSV file format is submitted to Kaltura server and Kaltura will download all of the videos listed in the CSV file.

B.2 Hardware requirenment

The hardware requirenments for video and application are as follows: - 3.0 GHz+ CPU (preferably Multi-core Intel Based)
- 4 GB+ of RAM

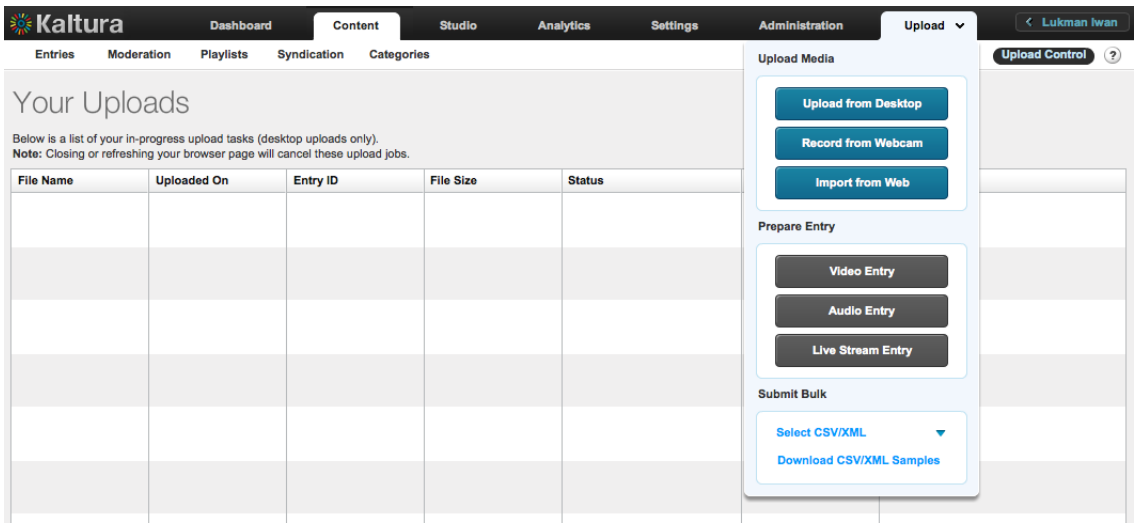


Figure B.1: Kaltura video upload interface

Title	Description	Tag	Media Type	URL
Melbourne Blue Show 2002	Melbourne 2002, Town Hall, 24 July	Circus Oz, Perfromance, 2002, Melbourne	video	http://kaltura.eres.rmit.edu.au/0_ucfg5d0t
New York 2001	New York, USA, 2001, New Victory Theater - 1 June	Circus Oz, Perfromance, 2001, Newyork	video	http://kaltura.eres.rmit.edu.au/0_ucvt4dsdt
...	...	...	...	...

Table B.1: Example of Kaltura video upload CSV file

- Current Storage Size : /data/ = 19T, /data2/ = 4.4T
- 1 static IP public

B.3 Software spesifications

Software spesifications for video and application servers are as follows:
- Operating System : Linux
- Web Server : Apache 2.2 or higher
- Database : MySQL 5.1.37 or higher
- Web Programming : PHP 5.3
- Mail Server : Sendmail, postfix
- Others modules : curl, memchached, ImageMagick, JRE 1.6, Pentaho, Xymon/Hobbit

B.4 Video server installation script

This installation scipt is based on RedHat server OS. Here is step on configuring video server:

1. Install webserver
   yum install httpd
   chkconfig httpd on

2. Set firewall
   -A INPUT -m state –state NEW -m tcp -p tcp –dport 22 -j ACCEPT
   -A INPUT -m state –state NEW -m tcp -p tcp –dport 80 -j ACCEPT
   -A INPUT -m state –state NEW -m tcp -p tcp –dport 443 -j ACCEPT

3. Instal and configure MySQL database yum install mysql mysql-server
   /usr/bin/mysqladmin -u root password '****'
   /usr/bin/mysqladmin -u root -h unixnpeap01.eres.rmit.edu.au password '****
   chkconfig mysqld on

4. Install PHP and configure PHP modules yum install php php-mysql
   yum install php-gd

yum install php-pecl-memcache.x86.64

yum install php-xsl

yum install php-pecl-apc

yum install php-imap

yum install php-devel

yum install libzip

5. Install PHPMyAdmin

6. Install required modules for Kaltura server yum install memcached

   chkconfig memcached on

   yum install java-1.6.0-openjdk.x86_64

   yum install imageMagick

   yum install libstdc++.so.6

   yum install compat-lib*

   yum install xulrunner.i686

7. Install Kaltura Commudity Edition Server, refers to http://ww.kaltura.org/

8. Install and configure Circus Oz application

Appendix C

Circus Oz Video Application

The Circus Oz application is intended to be accessed through the Internet and targeted those users who can access the application. It is designed for desktop computers although it can be run on mobile devices such as mobile phones and tablets. Given these factors, we developed Circus Oz as a web-based application that users can access via the user browser on their devices. Users do not need to install anything at their end; they just need a computer with an Internet browser application.

We developed the Circus Oz application using the LAMP (Linux, Apache, MySQL) framework. Specifically, the Circus Oz application is built as a web-based interface using the following software: Linux OS, Apache web server, MySQL database and PHP as programming language. Also, we use HTML and java script to manipulate the page.

C.1 Video data flow

We design the video data flow so that it begins with the video upload and ends with the video being stored on the Circus Oz application. The data flow of the Circus Oz video on this system architecture can be seen in Figure 3.6. The video processing flow is as follows:

1. Pre-processing: the original Circus Oz video is converted into a web video version size and watermarked with a Circus Oz logo.

2. Video upload: the Circus Oz video is uploaded from the desktop to the video server

through its web-based application. In this stage, the video data is also submitted to the Circus Oz database and includes the video id, title, date, and location.

3. Video server: the video application sends the video file to the video server through its API. The video data keys, i.e. video id and title, are also sent to the video server database. The video server transcodes the original video into two different formats: high and low resolutions.

4. Video storage: the video server saves the original and the transcoded video files in the storage server that is connected through the NFS protocol.

Once the video has been uploaded, the user can then retrieve the video through the following process: firstly, the user queries the video through the Circus Oz application. That user query is then searched against the Circus Oz database in order to obtain the video id of that user query. After that, using the video server API, the application asks the video server to deliver that video to the user. Finally, the video server retrieves that video from the video storage and delivers it to the user.

C.2 Video application features

Regarding the video server, here we include a list of the main functions of the Circus Oz application including the video upload, generation of image thumbnail, and retrieval of the video file. Firstly, the video can be uploaded through the Circus Oz application web interface as shown in Figure C.1. After that, the video metadata is entered (title, date, location, etc.) and saved on the Circus Oz database. In the next step, the Circus Oz application contacts the video server through its API to send and save the video file including some video metadata.

Next, through the video servers API, the image thumbnail can be generated from a specific video. This image can be produced from different locations on the video. A customized image size with specific image dimensions can also be generated. This image thumbnail is used by the application to display the search result and show a quick preview the video.

Lastly, once the user clicks play on the video screen, the Circus Oz application will contact the video server with the video id as a parameter. We can select a specific video

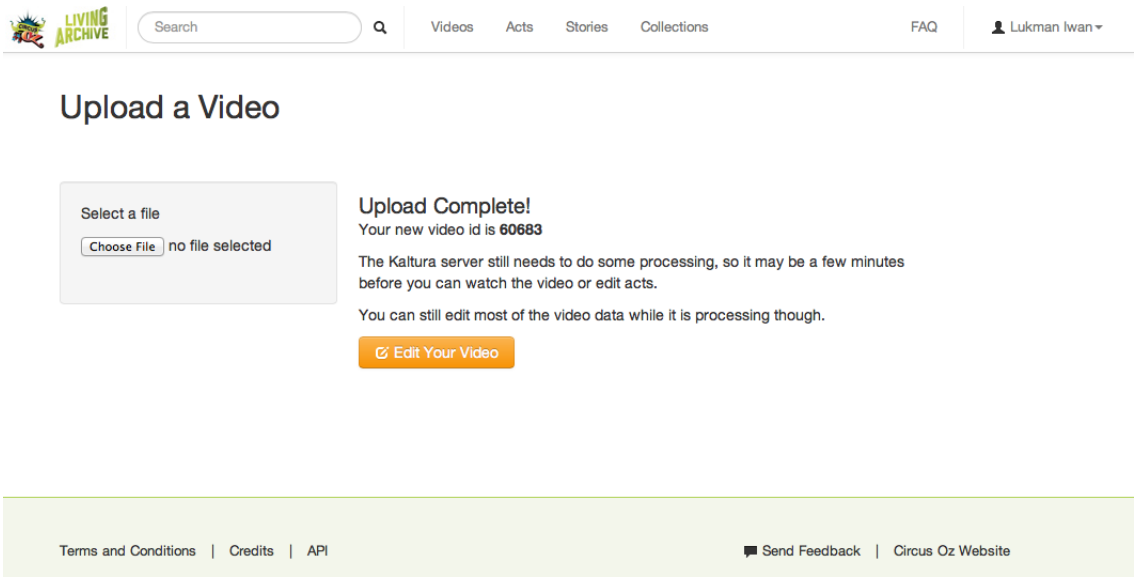


Figure C.1: Circus Oz application video upload interface

version available through the video server. By default, it loads a video source version.

C.3 Video application interfaces

The Circus Oz application is a web-based application that provides an interface with which users can interact for activities such as searching, browsing, tagging and watching videos. We structure the web- interface as follows:

1. Home page: this page is the landing page that the user will see when accessing: http://archive.circusoz.com/. The home page mainly displays a preview of clips including: act, video, and performers clips. Each clip is previewed with few image thumbnail animations representing that clip. The idea is that the user will see a summary of video content before deciding to watch that clip. Also, the user can browse the clips using the browsing menus for videos, acts, stories, and collections menus. The search box is provided on the home page enabling users to type a query related to circus data. While the registered user can login to the home page, the un-registered user can sign up to an account or simply watch the video without logging in.

2. Video page: this page is for browsing available videos that are sorted in chronological

order or show a date. Each video has basic metadata including: video name, acts, performers, and stories. Each basic metadata has a link to the related clips. For example, when user clicks on the act name, the page will display the video related to that act.

3. Act page:: the Act page displays all of the act names taken from all available Circus Oz videos. Once we click on the act name, it will take us to the video clip of that act.

4. Story page: the Story page is a list of users comments on a particular clip. The stories are made by registered users who post their comments on a particular clip.

5. Collection page: the Collection is actually a list of clips that are collected or grouped by registered users for a specific reason. The collection can be the acts, videos, or even the stories. One of the reasons for creating a collection is that an act may have the same act name on a different video. Another reason is that the list of acts may have the same music sound track. Users can set their collection as a public or private collection.

6. Search page: Search is an important component of this application as users generally want to know the contents of the Circus Oz collection based on their particular interest. For example, a user may want to know about a show performed in Melbourne in 1997 or may just want to search for juggling acts. Once the user has submitted his/her search query, the search result will be displayed as a list of videos that are related to that query. Initially, this list is ordered by relevance and show data. The search result is also grouped according to all elements of the performance: acts, stories and video type, documentaries, other, performances, TV commercials, and rehearsals.

Appendix D

Database Schema

D.1 Existing data

The existing Circus Oz data includes: show reports, human resources, and catalogue data. These data come from several different formats including: FileMaker database, spreadsheets, and text documents.

First is the show reports that contain the following fields:

- Location
- Date
- Session no
- Number of audience
- Building capacity
- Start time of the show
- Running order
- Person who in charge on that show
- Technical data: sound, lighting

- Costume data
- Comments

However, this show report is in a spreadsheet form file and it is not a structured database. As this is important data to begin with, we develop the script so that the report can be imported into the database.

Next is the Circus Oz human resources (HR) data that contain the following tables:

- season
- show
- personnel
- personnel data

The season table contains data about the Circus Oz season that is, the number of events for a year. As a show season could last for three to six months, there will be two or four seasons a year. Season has start date and end date. The show table contains data about the Circus Oz performances. One season can have a number of shows, and a season must have at least one show. The personnel table contains the personnel involved in a season and includes performers, manager, and backstage staff. The personnel data table has a date field that contains performers working date during that season and also their roles. A person can have two roles during a season: primary and secondary roles. For example, a person can have a role as a musician but at the same time s/he may be a performer. Finally, the personnel data table contains the personal details of each person. This personnel table is linked to Personal Data with person id.

In the HR database relation, there are three primary keys: season id, and person id. Using season id, we can generate the people who worked during a particular season and the list of shows for that season. However, as the Circus Oz collection focuses on the shows rather than the season, we need to obtain show data based on that HR database relationship. Fortunately, there is a season id field on both the show table and the personnel data table. Therefore, we can generate the data about anyone who performed in a particular show.

Last, another existing data is video catalogue and tape description. They are presented in spreadsheet and text documents. These documents are generally created when converting the video from analog to digital format. During the conversion process, the analog videotapes were labeled and recorded in a catalogue file that contains the following data:

- Tape id
- Location : location of show
- Date: show date
- Title: title appear on label
- Additional info: information regarding digitalization
- Type: show/rehearsal/promos/tva/tvc
- Format: original format: VHS, Betamax, UV
- Date migrated: the date when it migrated

The format of the tape id is XXX99.9.9. The first one to three digits represent the video type which in turn can be divided into 6 types: SH for show, M for miscellaneous, R for rehearsal, TVA for TV appearances, P for promos, and TVC for TV commercial. The following four digits represent the year, serial number of the show and tape number. For example, SH98.2.1 means that this video type is a show recorded in 1998; it is show number 2, and type number 1.

Based on the existing data, the table relations can be designed and shown in Figure D.1. The solid lines indicate that there is a data relationship between two or more tables. The broken lines indicate that there is no relation but the data in both tables are related. As can be seen from this figure, the database relation taken from the HR data is shown clearly. The primary keys are: season_id and person_id. In addition, although the video catalogue and tape desc is not presented in database form, there is potential key connection: tape_id.

There is no connection between the video catalogue and the session table. However, we can create the connection based on the date field in both tables. The connection is that if the

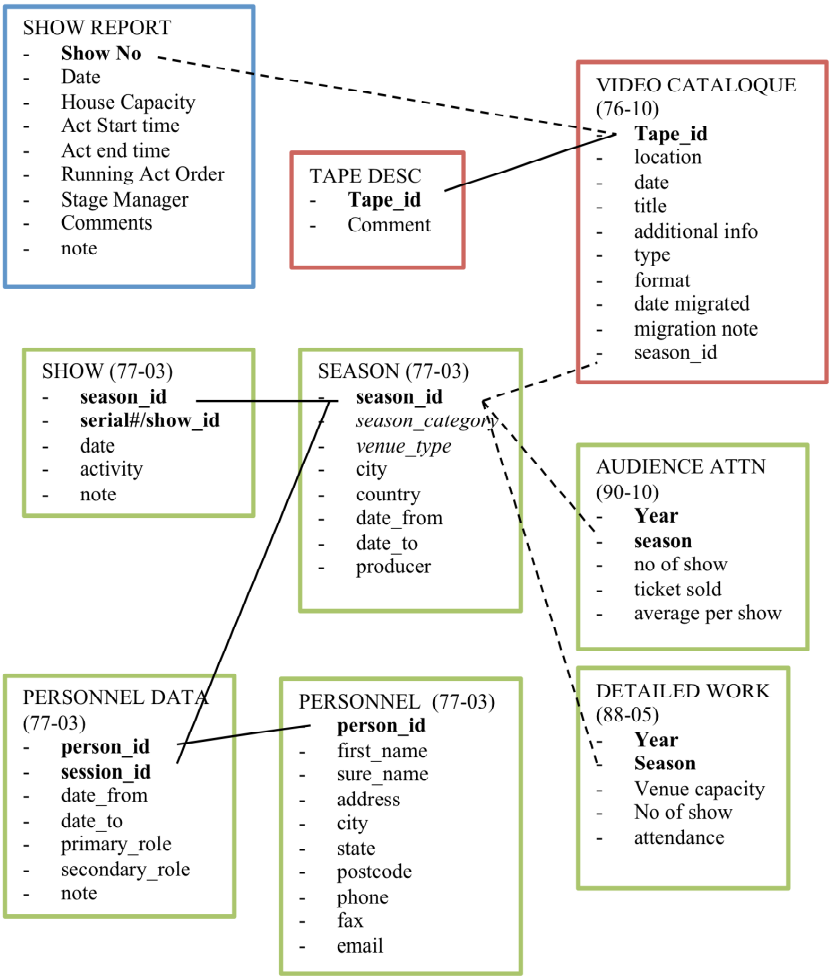


Figure D.1: Existing old Circus Oz database structure

date on the video catalogue table within the range of date on the session table, we can then populate the session id in the catalogue table. Hence, we can generate information about the performers who appear on the video.

The remaining issue regarding the database relation is the show report. Although the show number and the Tape_id are the same thing, they have different formats. Hence, we can have a list of shows in the video up to this stage.

D.2 Database schema prototypes

For this prototype, we designed the database based on video and clip structure. Video is a main table while clip is a secondary table. In the main table, a video is always recorded in one session. However, in the secondary table, this video can have multiple clips including: act, comment, images, etc. This video-clip structure is quite flexible so it can accommodate any additional clip type in the future.

Based on the above data, we conduct further analysis in order to develop database schema prototypes. We have implemented three different database schemas as shown in Figure D.2, Figure D.3, and Figure D.4. The first database schema (Figure D.2) is built to accommodate the existing old database for data populating as much as possible. There are 5 tables: video_catalogue, season, personnel, and performerse data. Video catalogue is the main table which contain list of video. It has 2 primary keys: one is tape_id which connected with tape_id at act table. This connection is to show list of act on that video. Another key is session_id. This key is to pull session metadata and performers who perform in that video through performers data and personnel data. All data from old database can be pulled to the new first database prototype except for show report.

The second database schema (Figure D.3) migrates the session data fields from session table to video table. The reason is that those fields are directly related to the video data. Since we are not actually revealing the session details to the users, we remove the session table and put the data directly into the video catalogue table. However, we still need the performers data that is connected through the session table. Hence, using a session id relation, we directly put the session id into the performers data. So at this stage, we are no longer

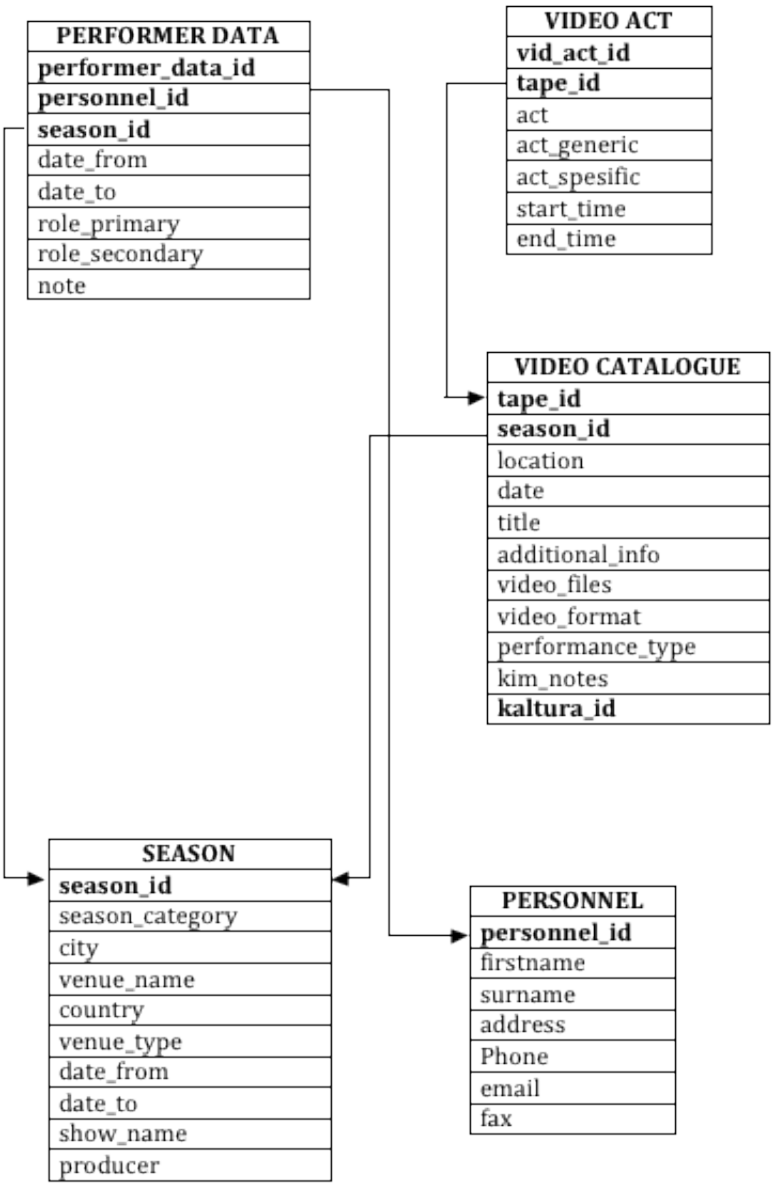


Figure D.2: First database prototype

depending on the session table.

In the third database schema (Figure D.4), clip and collection concepts are introduced. The clip table is related directly to the video table as a parent-child relation, whereas the collection table is related to video table through the clip table. In the previous database schema, we actually do not have a feature allowing a particular act video to be watched. Instead, we watch the act through the video that has a list of acts related to that video.

In this schema, the Circus Oz database stores video metadata and related information about Circus Oz such as: video catalogues, show reports, season details, personnel data and performers data. There are eight tables in the Circus Oz database including: video catalogue, season, personnel data, personnel, performers data, video act, act and video tag. These tables are connected with each other through the following primary keys: tape_id, season_id, personnel_id, personnel_data_id, performer_data_id, vid_act_id, act_id and vid_tag_id.

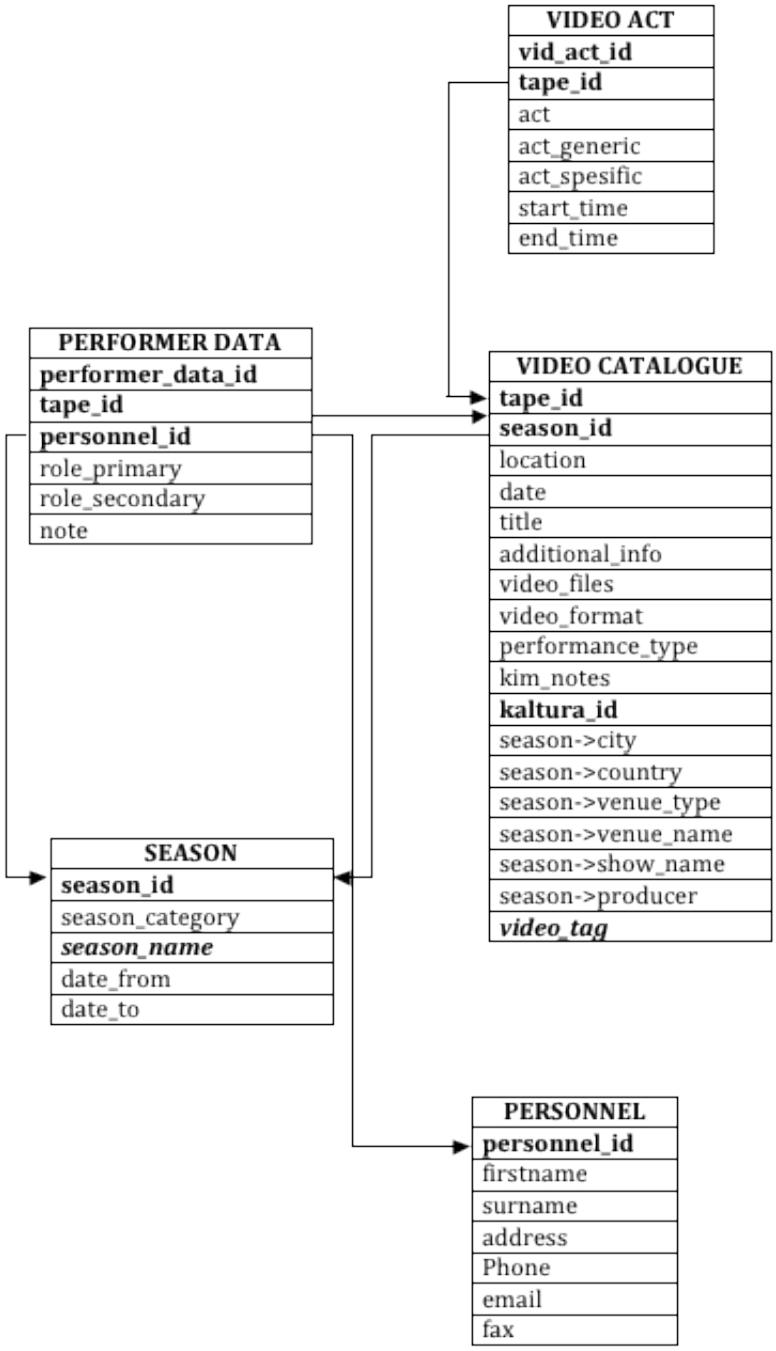


Figure D.3: Second database prototype

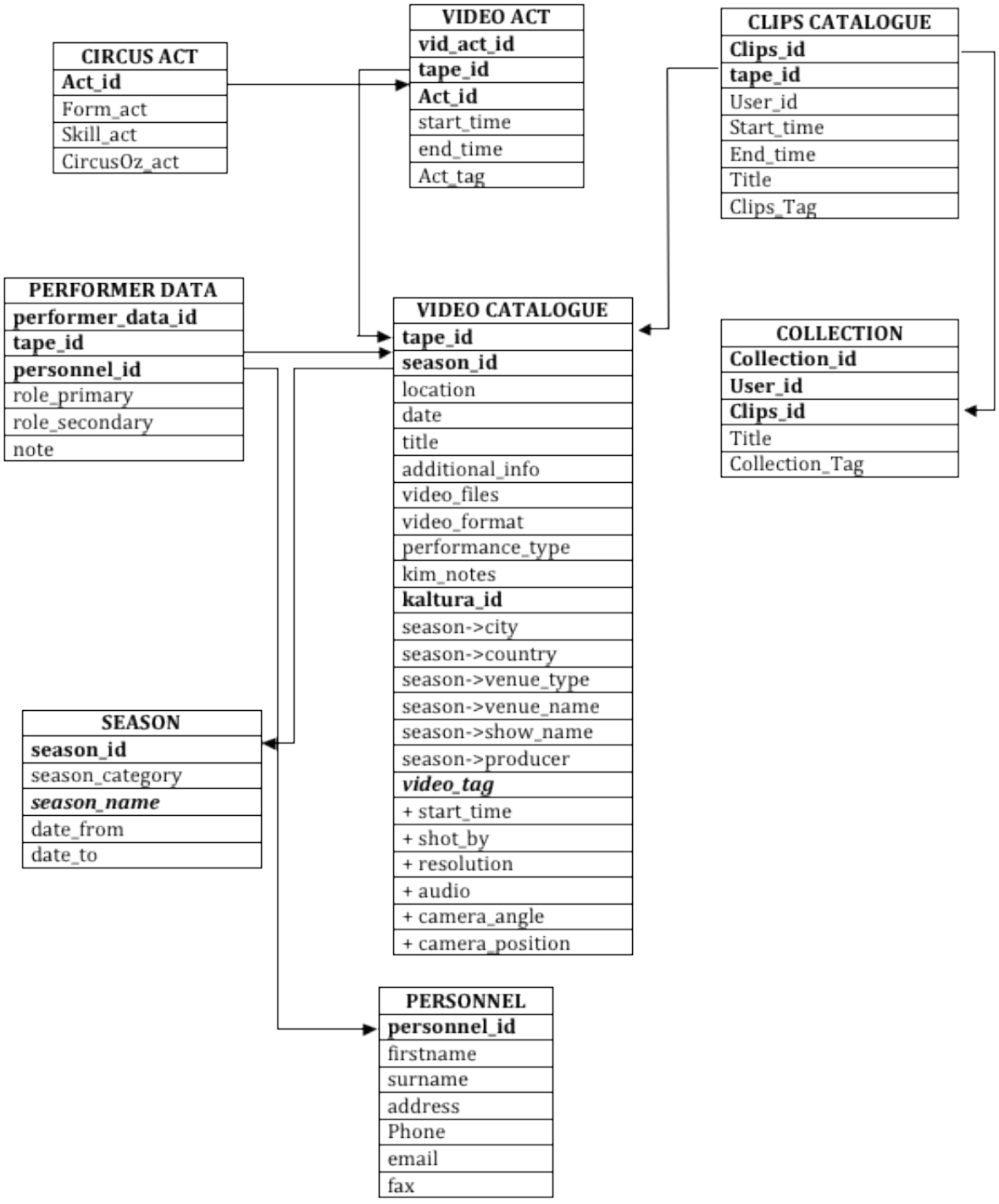


Figure D.4: Third database prototype

Appendix E

Search Functionality

One of the main features of the Circus Oz application is its search functionality. We have implemented three search engines: Kaltura built-in search, full-text search with MySQL and Sphinx search server.

E.1 Kaltura built-in searches

In the first prototype, we have implemented a Kaltura built-in search function. This built-in search function uses a Sphinxsearch server (http://sphinxsearch.com). On implementation, all data from Circus Oz database is copied into the Kaltura metadata. We created a customized Kaltura metadata to accommodate all data from the Circus Oz database. Through Kalturas API, we can perform a full-text search on all available metadata on the Kaltura.

However, it is hard to maintain all metadata in a customized Kaltura video server. The main reason is that there is an issue in terms of managing the synchronizing of data between the video server and the Circus Oz database. Whenever the database is updated in the Circus Oz database, the metadata in the video server has to be updated as well and vice versa. Also, changing the database schema makes it harder to maintain the data structure of both Circus Oz database and video server. Furthermore, the video server can accommodate only the video metadata while the clips metadata cannot be accommodated in the video server.

E.2 MySQL full-text searches

As an alternative solution, we have implemented the MySQL full-text search functionality. The reason is that all Circus Oz data is stored in a MySQL database. Hence, we need only one search server, that is MySQL. There are three MySQL search modes:

1. Natural Language

   The natural language matches a query against a text collection. The text collection could be from one or more fields on different tables. For example, a query Melbourne will match set pre-defined fields: video id, title, desc, place, and date. This search function will return a similarity measure between the query and text collection. It is not case sensitive and is sorted by relevance value. Relevance value is computed based on the number of words in the row, the number of unique words in that row, the total number of words in the collection, and the number of documents (rows) that contain a particular word. For example, the world aaabbb=1 word but aaabbb= 2 words. The beginning and the end of a word are determined by: space, comma, and period. Any word with less than four characters will be ignored. Although this can be set up, it will slow down the search performance. Every correct word in the collection and in the query is weighted according to its significance in the collection or query. Consequently, a word that is present in many documents has a lower weight (and may even have a zero weight), because it has lower semantic value in this particular collection. Conversely, if the word is rare, it receives a higher weight. The weights of the words are combined to compute the relevance of the row. The words that occur 50

2. Boolean

   Similar to the natural language query but 50

3. Query Expansion

   If we search the database, then the result is not only the database, but also mysql, oracle, db2, etc. Automatic relevance feedback works by performing the search twice where the second search concatenates the original query with the relevant result from the first one.

Of the three types of MySQL full-text search modes, the natural language search is preferable. The user that accesses the Circus Oz application is likely to be a public user who may not be familiar with the Boolean function. We have implemented the natural language search type on the Circus Oz application with the following sql query syntax:

select clip_type, title, description, tags, video_type, city, country, venue_type, venue_name, metadata, MATCH(clip_type, title, description, tags, video_type, city, country, venue_type, venue_name, metadata) AGAINST ("Melbourne") as score from search WHERE MATCH(clip_type, title, description, tags, video_type, city, country, venue_type, venue_name, metadata) AGAINST ("Melbourne").

However, the MySQL full-text search has the following issues:

1. Ranking differently from the actual result
2. Cannot search less than four characters. For example: fun, uk, usa, etc.
3. Cannot search frequent words (more than 50

E.3 Sphinx search engine

Sphinx[1] is an open source search server. When implementing the Sphinx search server, the following need to be considered: search index, search table content, match mode, field weight and advanced search.

Search Index

We index following fields:

1. Id
2. Clip_id
3. Clip_type
4. Title
5. Description

[1] http://sphinxsearch.com

6. Admin Tags
7. User Tags
8. Video_type
9. City
10. Country
11. Venue_type
12. Venue_name
13. Date
14. Video Id
15. Person
16. Username

Search table content

1. id, clip_id : key
2. video_type : performance, tv_appearance, documentary, other, rehearsal, promo, tv_commercial
3. Clip_type : video, act, comment
4. Admin tags: authoritative tag by admin
5. User tags: community tag by users
6. City, country, venue_type, venue_name, date, video_id : metadata taken from video table.
7. Clip_type = video
   - title : video_title
   - description : video_title
8. Clip_type = act
   - title : act_name
   - description : no_description

9. Clip_type = comment
   - title : i was there or i wasnt́ there but
   - description : comment

Match Mode

For example, a query: “Melbourne 2002”

1. Match_all : “Melbourne 2002” or “2002 Melbourne”
2. Match_any : “Melbourne 2002” or “2002 Melbourne” or “Melbourne” or “2002”
3. Match_Phrase : “Melbourne 2002”
4. Match_Boolean : , —, !, (...). “Melbourne 2002”
5. Match_Extended : , —, !, (...). “Melbourne 2002”
6. Field Search:(@username tim coldwell)—(@person tim coldwell)

Field Weight

1. By default all fields have the same weight = 1
2. Title has more weight than description
3. Person and title have same weight
4. Username has same weight with description
5. Tag, city, country, venue, date have higher weight than description and username but they have lower weight than title and person.
6. Act Categories
7. The field weight table is shown in Table E.1.

Advance Search

Search by particular field. For example: (@title Melbourne) — (@title Sydney) (@person Sosina) (@date 2002)

No	Field Name	Weight
1	Title	5
2	Person	5
3	Video_id	5
4	Admin Tag	4
5	Date	4
6	City	4
7	Country	4
8	Venue Type	4
9	Venue Name	4
10	User Tag	3
11	Description	3
12	Username	3
13	Video Type	3
14	Clip Type	3

Table E.1: The field weight on Sphinx Search.

E.3.1 Sphinxsearch configuration

This section explains on how to configure Sphinx search on the server. Sphinx search is installed on Circus Oz application. The step on configureing Sphinx search is as follows:

1. Get the Sphinx Search file source

   The Sphinx Search file source can be obtained from http://sphinxsearch.com. After that, extract this file source on user home folder at Circus Oz application server.

2. Create Sphinx Search file configuration

   The Sphinx Search file configuration for Circus Oz application is as follows:

   source catalog

   {

   type = mysql

   sql_host = localhost

   sql_user = root

   sql_pass = ****

   sql_db = circusoz

sql_sock = /var/run/mysqld/mysqld.sock

sql_port = 3306

sql_query =

SELECT * FROM search ;

sql_attr_uint = clip_type

sql_attr_uint = video_type

sql_query_info = SELECT * FROM clips WHERE id=id

}

index catalog

{

source = catalog

path = /root/search/data/catalog

morphology= stem_en

min_word$_len$ = 3

min_prefix_len = 0

min_infix_len = 3

}

searchd

{

port =3312

log = /root/search/log/circusoz_searchd.log

query_log= /root/search/log/circusoz_query.log

pid_file = /root/search/log/circusoz_searchd.pid

} Save this file as oz_sphinx.conf in the user home folder.

3. Create Sphinx Search Indexer file

   The Sphinx Search indexer file is as follow:

   /usr/bin/curl http://54.79.77.211/search/build_look/

   /root/search/sphinx206/bin/indexer –config /root/search/oz_sphinx.conf –all –rotate

   Save this file as run.sh on user home folder.

4. Start Sphinx Search engine The syntax for starting Sphinx Search engine is as follow:

   ./sphinx206/bin/searchd -c oz_sphinx.conf

5. Create cron The cron job contained the Sphinx search command to index the search content.

   It runs at least every one hour.

   0 * * * * /bin/sh /root/search/run.sh

Appendix F

Video Processing

On Circus Oz video application, the video need to pre-processing before they are uploaded into the video server. The pre-processing is including: video watermark, mute sound, and video merging.

F.0.2 Video watermark

We use FFMPEG video platform to put watermark on Circus Oz video. The FFMPEG command to watermark video is as follow:

```
ffmpeg -i [input\_filename] -vf ‘‘movie=logo\_circus\_oz.png [watermark];
[in] [watermark] overlay=main\_w-60:main\_h-60 [out]’’ [output\_filename].m4v
```

F.0.3 Sound muting

The process of muting audio segment on video is as follows:

1. Configure MySQL database tunnel

   The MySQL database tunnel is needed as the mute sound script is running on Kaltura server while the Circus Oz database is running on application server. An autossh script is used for connecting with application server. The MySQL database tunnel command is as follow:

```
autossh -M 0 -q -f -N -o ‘‘ServerAliveInterval 60’’ -o ‘‘ServerAliveCountMax 3’’
-L 3307:127.0.0.1:3306 lukman@10.118.97.2
```

2. Call mute_sound URL

   The mute_sound script can be run through accessing followingURL:

   ```
   http://kaltura.eres.rmit.edu.au/buffer/mute\_sound\?video\_id=[video\_id].
   ```

   This URL will return three possible values: 'ok', 'wait', and 'error'. The return value 'ok' means the mute sound processing will be run. The return value 'wait' means the mute sound processing will placed in a queue as other mute sound processing is still running. The return value 'error' means that the mute sound is failed. The process is failed because there is a problem with database connection or the video file is not found on the server.

3. Run mute_sound script

   Once the mute_sound url triggered, the mute_sound script will be running. The process of mute sound is as follow: First, mute_start and mute_end of muting sound will be taken from database. After that, the video file will be splitted into three segment. They are: start_video - mute_start, mute_start to mute_end, and mute_end to end_video. Next the audio on the second segment, mute_start to mute_end, is then removed with ffmpeg as follow:

   ```
   ffmpeg -i [input\_file] -vol 0 -ss 14.0 -t 0.999 [output\_file]
   ffmpeg -i [input.mp4] -vcodec copy -an [output.mp4]
   ```

   Finally, the three segments will be merged back into one file.

F.0.4 Video merging

The ffmpeg command for merging video is as follow:

```
-ffmpeg -i input1.avi -qscale:v 1 intermediate1.mpg
```

```
-ffmpeg -i input2.avi -qscale:v 1 intermediate2.mpg
-cat intermediate1.mpg intermediate2.mpg $>$ intermediate\_all.mpg
-ffmpeg -i intermediate\_all.mpg -qscale:v 2 output.avi
```

Appendix G

System Migration

System migration is migrating all components Circus Oz system from one server to another server. The components are including: video server, database, video storage and application server.

The current video server is using Kaltura community edition version 6.0. Other version of Kaltura should be fine as long as it has the same API functionality with Kaltura CE version 6.0.

G.1 Video migration

On migrating video from the current system to the new system environment, the above system requirements must be met. The steps on migrating video are follows:

1. Install video server: Kaltura Community Edition video server. Make sure kaltura is up and running.

2. Upload video

   Transferring video from one Kaltura server into other Kaltura server can be done as follow:

   (a) Get the kaltura_id from videos table on circusoz database. Spesifically, the following field need to export into csv file: video_table, title, year, city, country, video_type, and kaltura_id.

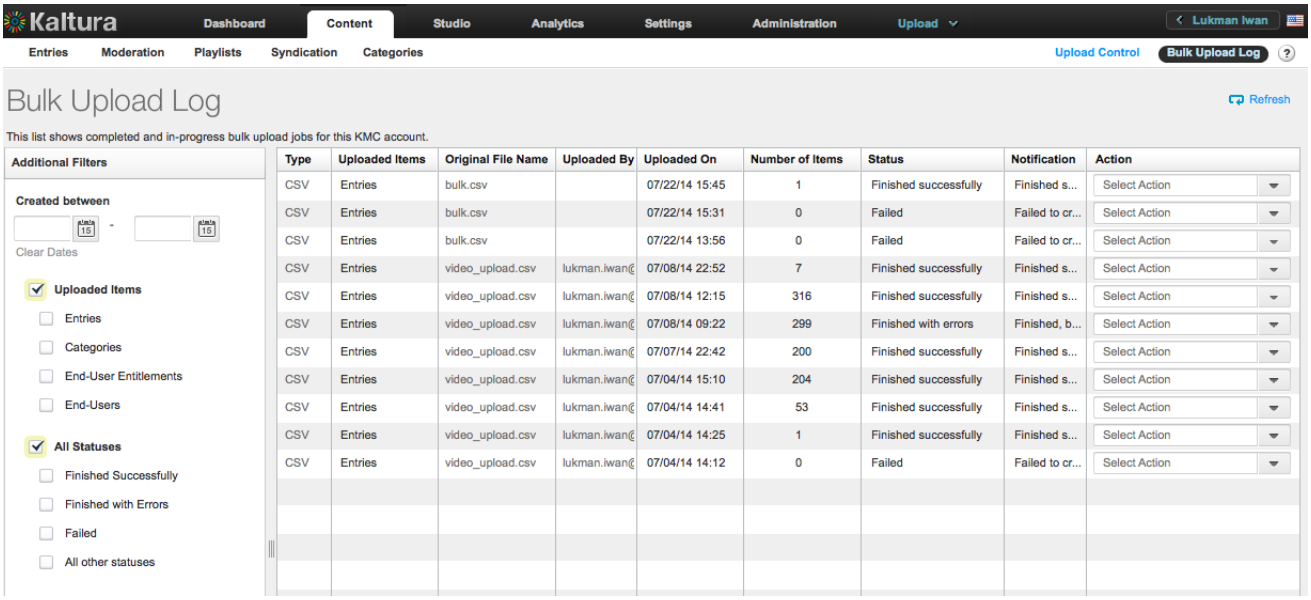


Figure G.1: Bulk upload progress interface

(b) Create a CSV file as following format:

Title: Title of video.

Description: city, country, video_type.

Tag: Video tags

Media type: 'video'

URL: a unique URL of each video that can be accessed through http protocol.

3. Submit the CSV file into Kaltura server The list of video listed in CSV file will be uploaded to the new Kaltura video server. Progress of uploading process can be traced in Kaltura KMC (Figure G.1). Kaltura will generate upload result once it completed. The upload result is CSV file contained new kaltura_id for each video record. If one of video record failed to upload, new kaltura_id will not be found. It will be showed an error message instead.

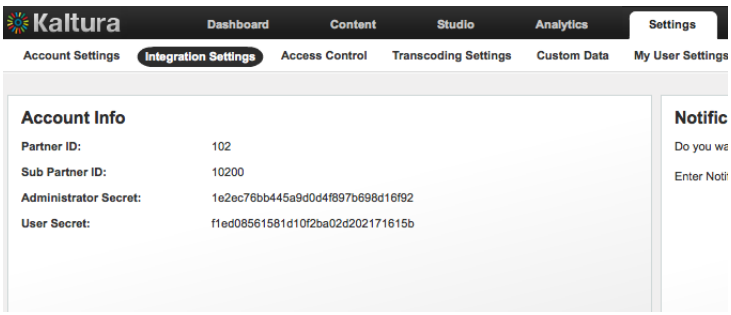
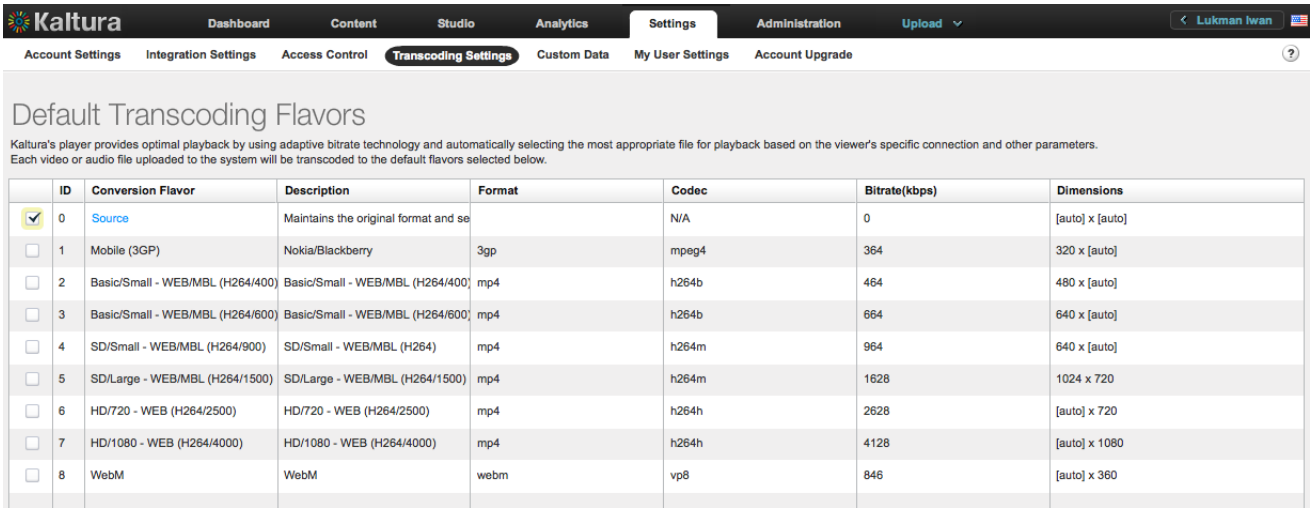


Figure G.2: Kaltura KMC interface

G.2 Circus Oz application configuration

The instruction on setting up Circus Oz web-based application on a new server is as follows:

1. Copy circusoz database from old server to the new server
2. Update kaltura_id on videos table with the new kaltura_id from new Kaltura server.
3. Copy all circusoz web-based application files into new server. Put these files into html root folder on the new server.
4. Setup Virtualhost
5. Change database configuration. Database configuration can be found in this file application/config/database.php. On that file, change the mysql account detail: host, usernama, and password.
6. Change Kaltura server URL. For example, the old Kaltura server URL: unixnpeap240.eres.rmit.edu.au. The new Kaltura server URL: monsoon.coremind.com.au. The file contained Kaltura url is: ./public/js/application/config.js and ./application/config/local/circusoz.php
7. Change the kaltura partner ID. For example from 104 on old server to 102 on new server. The kaltura partner ID number can be found in Kaltura KMC application on Setting - Integration Settings menu (Figure G.2). The file contained agent no is: ./public/js/application/config.js and ./application/config/local/circusoz.php.



	ID	Conversion Flavor	Description	Format	Codec	Bitrate(kbps)	Dimensions
☑	0	Source	Maintains the original format and se		N/A	0	[auto] x [auto]
☐	1	Mobile (3GP)	Nokia/Blackberry	3gp	mpeg4	364	320 x [auto]
☐	2	Basic/Small - WEB/MBL (H264/400)	Basic/Small - WEB/MBL (H264/400)	mp4	h264b	464	480 x [auto]
☐	3	Basic/Small - WEB/MBL (H264/600)	Basic/Small - WEB/MBL (H264/600)	mp4	h264b	664	640 x [auto]
☐	4	SD/Small - WEB/MBL (H264/900)	SD/Small - WEB/MBL (H264)	mp4	h264m	964	640 x [auto]
☐	5	SD/Large - WEB/MBL (H264/1500)	SD/Large - WEB/MBL (H264/1500)	mp4	h264m	1628	1024 x 720
☐	6	HD/720 - WEB (H264/2500)	HD/720 - WEB (H264/2500)	mp4	h264h	2628	[auto] x 720
☐	7	HD/1080 - WEB (H264/4000)	HD/1080 - WEB (H264/4000)	mp4	h264h	4128	[auto] x 1080
☐	8	WebM	WebM	webm	vp8	846	[auto] x 360

Figure G.3: Default transcoding flavors interface

8. Change secret code. The secret code is a secure connection Kaltura API code. This code is required to communicate with Kaltura Server. Th secret code can be found at Kaltura KMC page (Figure G.2). The file contained Kaltura secret code is: ./public/js/application/config.js and ./application/config/local/circusoz.php.

9. Change default transcoding falvors into source only (Figure G.3)

Bibliography

G. D. Abowd, M. Gauger, and A. Lachenmann. The Family Video Archive: An Annotation and Browsing Environment for Home Movies. In *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*, MIR '03, pages 1–8, New York, NY, USA, 2003. ACM.

D. Arthur and S. Vassilvitskii. K-means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 978-0-898716-24-5.

C. A. Bhatt and M. S. Kankanhalli. Multimedia data mining: state of the art and challenges. *Multimedia Tools and Applications*, 51(1):35–76, 2011.

F. Briggs, R. Raich, and X. Z. Fern. Audio Classification of Bird Species: A Statistical Manifold Approach. In *Ninth IEEE International Conference on Data Mining (ICDM '09)*, pages 51–60, 2009.

L.-H. Cai, L. Lu, A. Hanjalic, and H.-J. Zhang. A flexible framework for key audio effects detection and auditory context inference. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(3):1026–1039, 2006.

R. Cai, L. Lu, H.-J. Zhang, and L.-H. Cai. Highlight sound effects detection in audio stream. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, volume 3, pages III–37–40 vol.3, 2003.

Y. Cao, W. Tavanapong, K. Kim, and J. Oh. Audio-assisted scene segmentation for story browsing. In *2nd International Conference on Image and Video Retrieval*, pages 446–455. Springer-Verlag, 2003.

M. Ceillet, C. Roisin, and J. Carrive. Multimedia applications for playing with digitized theater performances. *Multimedia Tools and Applications*, 73(3):1777–1793, 2014.

V. Chasanis, A. Kalogeratos, and A. Likas. Movie segmentation into scenes and chapters using locally weighted bag of visual words. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, New York, New York, USA, 2009. ACM Press.

L.-H. Chen, Y.-C. Lai, and H.-Y. Mark Liao. Movie scene segmentation using background information. *Pattern Recognition*, 41(3):1056–1065, Mar. 2008.

M. Covell, S. Baluja, and M. Fink. Advertisement Detection and Replacement using Acoustic and Visual Repetition. In *Multimedia Signal Processing, 2006 IEEE 8th Workshop on*, pages 461–466, 2006.

L.-Y. Duan, J. Wang, Y. Zheng, J. S. Jin, H. Lu, and C. Xu. Segmentation, categorization, and identification of commercial clips from TV streams using multimodal analysis. In *Proceedings of the 14th annual ACM international conference on Multimedia*, MULTIMEDIA '06, pages 201–210, New York, NY, USA, 2006. ACM.

L.-Y. Duan, J. Wang, Y.-T. Zheng, H. Lu, and J. S. Jin. Digesting Commercial Clips from TV Streams. *IEEE MultiMedia*, 15(1):28–41, 2008. ISSN 1070-986X.

O. Gillet and G. Richard. Comparing Audio and Video Segmentations for Music Videos Indexing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages V–V, 2006.

B. Günsel, A. M. Ferman, and A. M. Tekalp. Temporal Video Segmentation Using Unsupervised Clustering and Semantic Object Tracking. *Journal of Electronic Imaging*, 7:592–604, 1998.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Exporation Newsletter*, 11(1):10–18, 2009.

A. Hanjalic, R. Lagendijk, and J. Biemond. Automatically segmenting movies into logical story units. In *Proceedings of the 3rd International Conference on Visual Information and Information Systems*, pages 229–236. Springer-Verlag, 1999.

H. Harb and L. Chen. A General Audio Classifier Based on Human Perception Motivated Model. *Multimedia Tools and Applications*, 34(3):375–395, Sept. 2007.

R. Hjelsvold, S. Langørgen, R. Midstraum, and O. Sandstå. Integrated video archive tools. In *Proceedings of the Third ACM International Conference on Multimedia*, MULTIMEDIA '95, pages 283–293, New York, NY, USA, 1995. ACM.

I. Ide, H. Mo, N. Katayama, and S. Satoh. Topic Threading for Structuring a Large-Scale News Video Archive. In P. Enser, Y. Kompatsiaris, N. OConnor, A. Smeaton, and A. Smeulders, editors, *Image and Video Retrieval SE - 18*, volume 3115 of *Lecture Notes in Computer Science*, pages 123–131 LA – English. Springer Berlin Heidelberg, 2004.

I. Ide, T. Kinoshita, H. Mo, N. Katayama, and S. Satoh. trackThem: Exploring a Large-Scale News Video Archive by Tracking Human Relations. In G. Lee, A. Yamada, H. Meng, and S. Myaeng, editors, *Information Retrieval Technology SE - 42*, volume 3689 of *Lecture Notes in Computer Science*, pages 510–515. Springer Berlin Heidelberg, 2005.

L. Iwan. Clues for temporal segmentation of circus videos into acts. In D. Carlin and L. Vaughan, editors, *Performing Digital: Multiple Perspectives on a Living Archive*, chapter 10. Ashgate, Farnham UK, 2015.

L. H. Iwan and J. A. Thom. Temporal video segmentation: Detecting the end- of-act in Circus performance video archives. *Multimedia Tools and Applications*, (to appear, accepted for publication 1 December 2015).

R. Jarina and J. Olajec. Discriminative Feature Selection for Applause Sounds Detection. In *Image Analysis for Multimedia Interactive Services, 2007. WIAMIS '07. Eighth International Workshop on*, page 13, 2007.

S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967. ISSN 1860-0980.

E. Kijak, G. Gravier, L. Oisel, and P. Gros. Audiovisual Integration for Tennis Broadcast Structuring. *Multimedia Tools and Applications*, 30(3):289–311, Sept. 2006.

S. Kiranyaz, A. F. Qureshi, and M. Gabbouj. A generic audio classification and segmentation approach for multimedia indexing and retrieval. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3):1062–1081, 2006.

N. Lesser and D. P. W. Ellis. Clap detection and discrimination for rhythm therapy. In *Procedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 37–40, Philadelphia, Pennsylvania, 2005.

Y.-X. Li, Q.-H. He, S. Kwong, T. Li, and J.-C. Yang. Characteristics-based effective applause detection for meeting speech. *Signal Processing*, 89(8):1625–1633, 2009.

R. Lienhart, C. Kuhmunch, and W. Effelsberg. On the detection and recognition of television commercials. In *Multimedia Computing and Systems '97. Proceedings., IEEE International Conference on*, pages 509–516, 1997.

R. Lienhart, S. Pfeiffer, and W. Effelsberg. Scene determination based on video and audio features. In *IEEE International Conference on Multimedia Computing and Systems*, pages 685–690, Florence, 1999.

C. Liu, D. Wang, J. Zhu, and B. Zhang. Learning a Contextual Multi-thread Model for Movie/TV Scene Segmentation. *IEEE Transactions on Multimedia*, (c):1–1, 2012.

N. Liu, Y. Zhao, and Z. Zhu. Commercial recognition in TV streams using coarse-to-fine matching strategy. *In Multimedia Information Processing-PCM 2010*, pages 296–307, 2010.

N. Liu, Y. Zhao, Z. Zhu, and H. Lu. Exploiting Visual-Audio-Textual Characteristics for Automatic TV Commercial Block Detection and Segmentation. *Multimedia, IEEE Transactions on*, 13(5):961–973, 2011.

L. Lu, H.-J. Zhang, and S. Z. Li. Content-based audio classification and segmentation by using support vector machines. *Multimedia Systems*, 8(6):482–492, 2003.

C. Manoj, S. Magesh, A. S. Sankaran, and M. S. Manikandan. Novel approach for detecting applause in continuous meeting speech. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, volume 3, pages 182–186, 2011.

P. Marmaroli, A. Dufaux, and A. Delidais. Automatic Detection of Applause in the Montreux Jazz Festival Concerts. In *2nd Workshop on Standards and Technologies in Multimedia Archives and Records*, 2013.

D. McEnnis, C. McKay, I. Fujinaga, and P. Depalle. JAudio: a feature extraction library. *6th International Conference on Music Information Retrieval*, 2005.

C. McKay. Automatic music classification and similarity analysis. *Course paper, Universite de Montreal, Canada*, 2005.

C. McKay. *Automatic Music Classification with jMIR*. PhD thesis, McGill University, Canada, 2010.

H. Mo, F. Yamagishi, I. Ide, N. Katayama, S. Satoh, and M. Sakauchi. Key Image Extraction from a News Video Archive for Visualizing Its Semantic Structure. In *Proceedings of the 5th Pacific Rim Conference on Advances in Multimedia Information Processing - Volume Part I*, PCM'04, pages 650–657, Berlin, Heidelberg, 2004. Springer-Verlag.

K. N. Ngan and H. Li. *Video Segmentation and Its Applications*. Springer Publishing Company, Incorporated, 1st edition, 2011.

J. Olajec, R. Jarina, and M. Kuba. GA-Based Feature Extraction for Clapping Sound Detection. In *Neural Network Applications in Electrical Engineering, 2006. NEUREL 2006. 8th Seminar on*, pages 21–25, 2006.

D. Pelleg and A. W. Moore. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 727–734, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-707-2.

D. Petrelli and D. Auld. An examination of automatic video retrieval technology on access to the contents of an historical video archive. *Program*, 42(2):115–136, 2008.

J. Pinquier, J. Arias, and R. Andre-Obrecht. Audio classification by search of primary. In *International Workshop on Image, Video and Audio Retrieval and Mining*, Sherbrooke, 2004.

D. A. Sadlier, S. Marlow, N. E. O'Connor, and N. Murphy. Automatic TV Advertisement Detection from MPEG Bitstream. In *Proceedings of the 1st International Workshop on Pattern Recognition in Information Systems: In conjunction with ICEIS 2001*, PRIS '01, pages 14–25. ICEIS Press, 2001.

P. Sarala, V. Ishwar, A. Bellur, and H. A. Murthy. Applause identification and its relevance to archival of Carnatic music. In *Proceedings of the 2nd CompMusic Workshop*, Istanbul, Turkey, 2012.

Z. Shi, J. Han, and T. Zheng. Heterogeneous mixture models using sparse representation features for applause and laugh detection. In *Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on*, pages 1–5, 2011.

P. Sidiropoulos. Temporal video segmentation to scenes using high-level audiovisual features. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(8):1163–1177, Aug. 2011.

P. Silva. Classification, segmentation and chronological prediction of cinematic sound. *International Conference on Machine Learning and Applications*, 11, 2012.

S. W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, San Diego, CA, USA, 1997.

C. G. M. Snoek and M. Worring. Multimodal Video Indexing: A Review of the State-of-the-art. *Multimedia Tools and Application*, 25(1):5–35, Jan. 2005.

K. Subashini and S. Palanivel. Audio-video based Segmentation and Classification using SVM and AANN. *International Journal of Computer Applications*, 53(18):43–49, 2012.

K. Subashini, S. Palanivel, and V. Ramaligam. Combining audio-video based segmentation and classification using SVM. *International Journal of Signal System Control and Engineering Application*, 4:69–73, 2011.

H. Sundaram and C. Shih-Fu. Video scene segmentation using video and audio features. In *IEEE International Conference on Multimedia and Expo*, volume 2, pages 1145–1148 vol.2, 2000.

T. Theodorou, L. Mporas, and N. Fakotakis. Automatic sound classification of radio broadcast news. *International Journal of Signal Processing, Image Processing, and Pattern Recognition*, 5(1):37–47, 2012.

T. Trifonov and T. Georgieva. Client/server system for managing an audio and video archive for unique bulgarian bells. *WSEAS Trans. Info. Sci. and App.*, 6(4):660–669, Apr. 2009.

C. Uhle. Applause Sound Detection. *Journal Audio Engineering Society*, 59(4):213–224, 2011.

P. M. B. van Roosmalen, R. L. Lagendijk, and J. Biemond. Restoration and Storage of Film and Video Archive Material. In *NATO summer school*, 1998.

L. Vaughan, R. Stanton, L. Iwan, J. Yuille, J. Mullet, D. Carlin, J. Thom, and A. Miles. Multimodal experiments in the design of a living archive. In *Design Research Conference*, pages 650–657, Copenhagen, 2013.

J. Wakefield. Surveillance cameras to predict behaviour, 2002. URL http://news.bbc.co.uk/2/hi/sci/tech/1953770.stm.

J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang. A Formal Study of Shot Boundary Detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(2):168–186, 2007.

D. Zhang and D. Ellis. Detecting Sound Events in Basketball Video Archive. Technical report, Columbia University, 2001. URL http://www.ee.columbia.edu/~dpwe/classes/e6820-2001-01/projects/dqzhang.pdf.

T. Zhang and C.-C. J. Kuo. Hierarchical classification of audio data for archiving and retrieving. In *Proceedings of IEEE International Conference - the Acoustics, Speech, and Signal Processing*, volume 6, pages 3001–3004. IEEE Computer Society, 1999.

H. Zou, Q. Lu, J. C. Durack, C. Chao, C, H. Strasberg, Y. Zhang, M. Tsai, K.Melmon, and J. Hahn. Structured data management–the design and implementation of a web-based video archive prototype. In *Annual Symposium Proceedings Archive*, pages 786–790, 2001.